SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 1-14, February 1998

THE RING LOADING PROBLEM*

ALEXANDER SCHRIJVER†, PAUL SEYMOUR‡, AND PETER WINKLER§

Abstract. The following problem arose in the planning of optical communications networks which use bidirectional SONET rings. Traffic demands $d_{i,j}$ are given for each pair of nodes in an n-node ring; each demand must be routed one of the two possible ways around the ring. The object is to minimize the maximum load on the cycle, where the load of an edge is the sum of the demands routed through that edge.

We provide a fast, simple algorithm which achieves a load that is guaranteed to exceed the optimum by at most 3/2 times the maximum demand, and that performs even better in practice. En route we prove the following curious lemma: for any $x_1, \ldots, x_n \in [0, 1]$ there exist $y_1, \ldots, y_n$ such that for each k, $|y_k| = x_k$ and

$$\left| \sum_{i=1}^{k} y_i - \sum_{i=k+1}^{n} y_i \right| \leq 2.$$

Key words. SONET ring, load balancing, optical network design

AMS subject classifications. 90B10, 90C10, 94C15

PII. S0895480195294994

1. Introduction.

Around the world, billions of dollars are being spent by telephone operating companies to replace copper circuits with optical fiber, vastly increasing potential bandwidth and opening the network to multiple data-types—including video. The dominant technological standard in the United States is the Synchronous Optical NETwork (SONET) [1]. In one very popular configuration called a SONET ring, nodes (typically telephone central offices) are connected by a ring of fiber, with each node sending, receiving, and relaying messages by means of a device called an add-drop multiplexer (ADM).

SONET rings enjoy several advantages over other network configurations. The vertex-symmetry of the rings ensures that nodes play the same role and are similarly equipped, and the connectivity of the cycle protects against failure of either a link (that is, an edge) or a node. Thus, a major task of network-planning software, including Bellcore's SONET Toolkit™ [2], is to identify groups of nodes which can be turned into SONET rings in such a way as to satisfy traffic demands in a cost-efficient manner.

The capacity of a SONET ring varies from ring to ring but is the same for each link of a ring, and the cost of a ring (all other factors being equal) is an increasing function of its capacity. It is not the fiber itself but the ADMs which limit bandwidth. However, the effect is the same: for each SONET ring there is a capacity C such that no link of the ring may carry more than C units of traffic.

In some SONET rings all traffic is routed clockwise (unless a fault has occurred) and the capacity is selected so as to handle the sum of all the point-to-point demands

* Received by the editors October 6, 1995; accepted for publication (in revised form) October 9, 1996. The majority of the research described in this paper was conducted while the second and third authors were employed by Bellcore; the first was a visiting consultant.

http://www.siam.org/journals/sidma/11-1/29499.html

† Mathematics Center, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands (lex@cwi.nl).

‡ Department of Mathematics, Princeton University, Princeton, NJ 08544 (pds@math.princeton.edu).

§ Bell Labs 2C-379, 700 Mountain Avenue, Murray Hill, NJ 07974 (pw@lucent.com).

between nodes of the ring. Such "unidirectional" SONET rings will not concern us here.

In bidirectional rings, however, a routing is chosen independently for each pair of nodes, and all traffic between those nodes (in either direction) is sent by that route. Clearly, bidirectional rings are much more bandwidth-efficient; for example, when demands are uniform they can carry four times the traffic of a unidirectional ring having the same capacity.

In order to compute the capacity required for a proposed bidirectional SONET ring, the planning software must route the projected traffic demands in such a way as to minimize, or at least approximately minimize, the maximum load on any link. The problem is described formally below. We remark that the actual capacity selected for a proposed ring is further adjusted to allow for failures and abnormal demands, and that there is a discrete set of standard capacities from which to choose; but these considerations do not change the objective.

2. Notation and terminology. The problem is formally stated as follows:

RING LOADING

INSTANCE: Ring size n and nonnegative integers $d_{i,j}$, $1 \le i < j \le n$.
QUESTION: Find a map $\phi : \{(i,j) : 1 \le i < j \le n\} \longrightarrow \{0,1\}$ which minimizes $L = \max_{1 \le k \le n} L_k$, where

$$L_k = \sum \{d_{i,j} : \phi(i,j) = 1 \text{ and } k \in [i,j)\} + \sum \{d_{i,j} : \phi(i,j) = 0 \text{ and } k \notin [i,j)\}.$$

The notation "$[i,j)$" is used here for the half-closed integer interval $\{i, i+1, \ldots, j-1\}$.

To make RING LOADING a decision problem as in [7], we append a target value T to the instance and ask whether there is a ϕ for which $L \le T$.

Each $d_{i,j}$ is called a *demand*, and the map ϕ is called a *routing*. Setting $\phi(i,j) = 0$ amounts to routing the traffic between nodes i and j the "back" way, that is, through the link $\{n,1\}$. When $\phi(i,j) = 1$ we say that the (i,j)th demand has been routed through the "front."

The routing induces a *load* L_i on each link $\{i, i+1\}$, namely, the sum of the demands routed through that link. The largest load is the *ringload* L, the quantity to be minimized.

3. Theory and reality. The decision form of RING LOADING is clearly in the class NP since the routing provides a witness which is only $\binom{n}{2}$ bits long. In fact, RING LOADING is an integer multicommodity flow problem (the reader is referred to [4] for a survey on such problems); in general such problems are NP-complete, but we are dealing with a very special case.

Technically, the input size for an instance of RING LOADING is slightly more than

$$\lceil \log n \rceil + \lceil \log T \rceil + \sum_{1 \le i < j \le n} \lceil \log d_{i,j} \rceil,$$

relative to which RING LOADING is NP-complete. A simple reduction is available from the PARTITION problem [7, p. 223], in which positive integers $a_1, \ldots, a_m$ are given and the question is whether one can divide them into two groups of equal sum. Put $n = m + 3$, $d_{i,m+2} = a_i$ for $1 \le i \le m$, and $d_{m+1,m+2} = d_{m+2,m+3} = \sum a_i/2$. Set all other demands equal to zero, and let $T = \sum a_i$. Then a good routing must send

$d_{m+1,m+2}$ and $d_{m+2,m+3}$ the short way (front) and must partition the other demands so that $L_{m+1} = L_{m+2} = T$. This solves PARTITION, and vice versa.

An even easier reduction—with just two nodes—was given by Cosares and Saniee [3] and was made possible by their slightly more general RING LOADING formalization in which more than one demand per node pair is allowed. (The positive results to follow are also easily extended to cover the more general formulation; we prefer the more restrictive version for notational reasons.)

However, the reduction from PARTITION says nothing about the tractability of RING LOADING in practice, because PARTITION is solvable in time polynomial in $m \cdot \max a_i$ and actual demand sizes for RING LOADING are not large numbers. In fact, traffic demands are estimates to begin with, and the range 0 to 100 units is typically adequate. Thus, we may even take the maximum demand D to be bounded by a reasonable *constant*. The size n of a SONET ring is currently restricted to about 20. With these parameters, an instance of PARTITION can be solved using dynamic programming by hand!

Modest as the parameters are, however, they do not permit exhaustive search of the $2^{\binom{n}{2}}$ possible routings, and the PARTITION-to-RING LOADING reduction does not appear to permit reversal. As far as we know, any of the following three statements may be true (see [7] for descriptions of CLIQUE and CHROMATIC NUMBER):

- RING LOADING (like PARTITION) can be solved in time polynomial in n and D.
- RING LOADING (like CLIQUE) can be solved in time polynomial in n but only if a bound on the maximum demand D is fixed.
- RING LOADING (like CHROMATIC NUMBER) is NP-complete even for (some) fixed D.

Mercifully, the $D = 1$ case is solvable in time polynomial in n. The proof is due to Frank [5] and is explained nicely in [6]; it relies on a theorem of Okamura and Seymour [9]. This case is important because in some cases demands *can* be split, but only at integral values, and can thus be regarded as a multiplicity of unit demands. In fact, as we shall demonstrate, our approximation algorithm for RING LOADING actually solves this case exactly.

We do not have a fast exact algorithm, either in theory or in practice, for the RING LOADING problem with $D>1$. Fortunately, in practice, a reasonable approximate solution to RING LOADING was acceptable. There was no room for compromise on the issue of computation time: the RING LOADING problem had to be solved in a matter of seconds at most, because it was part of a frequently called subroutine for determining the cost of *proposed* SONET rings. The full program considers enormous numbers of potential SONET rings and is supposed to work on run-of-the-mill serial computers.

To be precise, we sought an algorithm A with the following three properties, listed in order of importance:

1. A must be fast.
2. A should provide a solution to RING LOADING which exceeds the optimum load by no more than about 5% in most cases.
3. A should, if possible, come with a performance guarantee for both (1) and (2).

As it turns out, these properties were obtainable with a fairly simple algorithm whose efficiency does not much depend on D (the demands can be treated as real numbers).

4. Linear relaxation. The "relaxed" version of RING LOADING, in which demands may be split (that is, sent partly around the front, partly around the back), is

formulated as follows:

RELAXED RING LOADING

INSTANCE: Ring size n and nonnegative integers $d_{i,j}$, $1 \le i < j \le n$.
QUESTION: Find a map $\phi^* : \{(i,j) : 1 \le i < j \le n\} \longrightarrow [0,1]$ which minimizes $L^* = \max_{1 \le k \le n} L_k^*$, where

$$L_k^* = \sum\{\phi^*(i,j)d_{i,j} : k \in [i,j)\} + \sum\{(1-\phi^*(i,j))d_{i,j} : k \notin [i,j)\}.$$

Since this is now a linear programming problem, it is solvable in polynomial time [8]. In fact, we shall see that a solution to RELAXED RING LOADING can be obtained in a very fast greedy fashion, even if we demand the additional property described in Proposition 4.1.

It is useful to think of demands geometrically as weighted chords in a circle representing the SONET ring. Two demands $d_{g,h}$ and $d_{i,j}$, with $g < h$ and $i < j$, are said to *cross* if all of the indices are distinct and if exactly one of i and j lies in (g,h); otherwise they are said to be *parallel*. In particular, demands such as $d_{i,j}$ and $d_{i,k}$, which share a node, are parallel.

A link which lies between two chords representing parallel demands is said to be "between" the demands. Finally, a routing ϕ^* for the RELAXED RING LOADING problem is said to *split* a demand $d_{i,j}$ if $0 < \phi^*(i,j) < 1$.

PROPOSITION 4.1. *Let ϕ^* be a routing for an instance of* RELAXED RING LOADING *which achieves the optimal load L^* and is also minimal, in the sense that no other routing has $L_i \le L_i^*$ for every i and $L_j < L_j^*$ for some j. Then no link which lies between two parallel demands will carry traffic from both demands.*

Proof. Assume otherwise, letting link $\{k, k+1\}$ carry a quantity a of traffic from demand $d_{g,h}$ and $b \ge a$ from $d_{i,j}$. After rerouting a quantity a of traffic from each demand so as to no longer pass through the kth link, no link suffers an increased load. This contradicts the minimality of ϕ^*. □

Proposition 4.1 fails for RING LOADING as can be seen from the example in Fig. 1, where $n = 8$ and the nonzero demands are $d_{2,3} = d_{1,4} = 1$ and $d_{6,7} = d_{5,8} = 2$. The optimal $\{0,1\}$-assignment sends both $d_{1,4}$ and $d_{5,8}$ the long way around the ring, achieving $L = 3$; no other assignment can do better than $L = 4$. What is significant, however, is that the proposition does hold in the case of $\{0,1\}$ demands.

We can turn RELAXED RING LOADING into a decision problem in a more general way than before. We append to the instance a capacity C_i for each link $\{i, i+1\}$ and ask whether there is a routing ϕ^* for which $L_i \le C_i$ for each i. In the following it will be useful to regard node labels as integers modulo n, so that, for example, the link $\{n,1\}$ is also written $\{n, n+1\}$ and the half-open interval $(g,h]$ is interpreted as $\{g, g+1, \ldots, n-1, n, 1, 2, \ldots, h-1\}$ if $h < g$.

Each pair of links $\{g, g+1\}$, $\{h, h+1\}$, with $g < h$, constitutes a *cut* of capacity $C_g + C_h$ in the network. We may think of a cut as a chord connecting the midpoints of the links $\{g, g+1\}$ and $\{h, h+1\}$; if a demand $d_{i,j}$ crosses this chord, any routing will contribute load $d_{i,j}$ to the cut's two links. Thus, if the instance is solvable, then $D_{g,h} \le C_g + C_h$, where

$$D_{g,h} := \sum\{d_{i,j} : i \le g \text{ and } j \in (g,h], \text{ or } i \in (g,h] \text{ and } j > h\}$$

is the total traffic demand across the cut. The following converse is a special case of the Okamura–Seymour theorem [9]; we give a simple proof here.

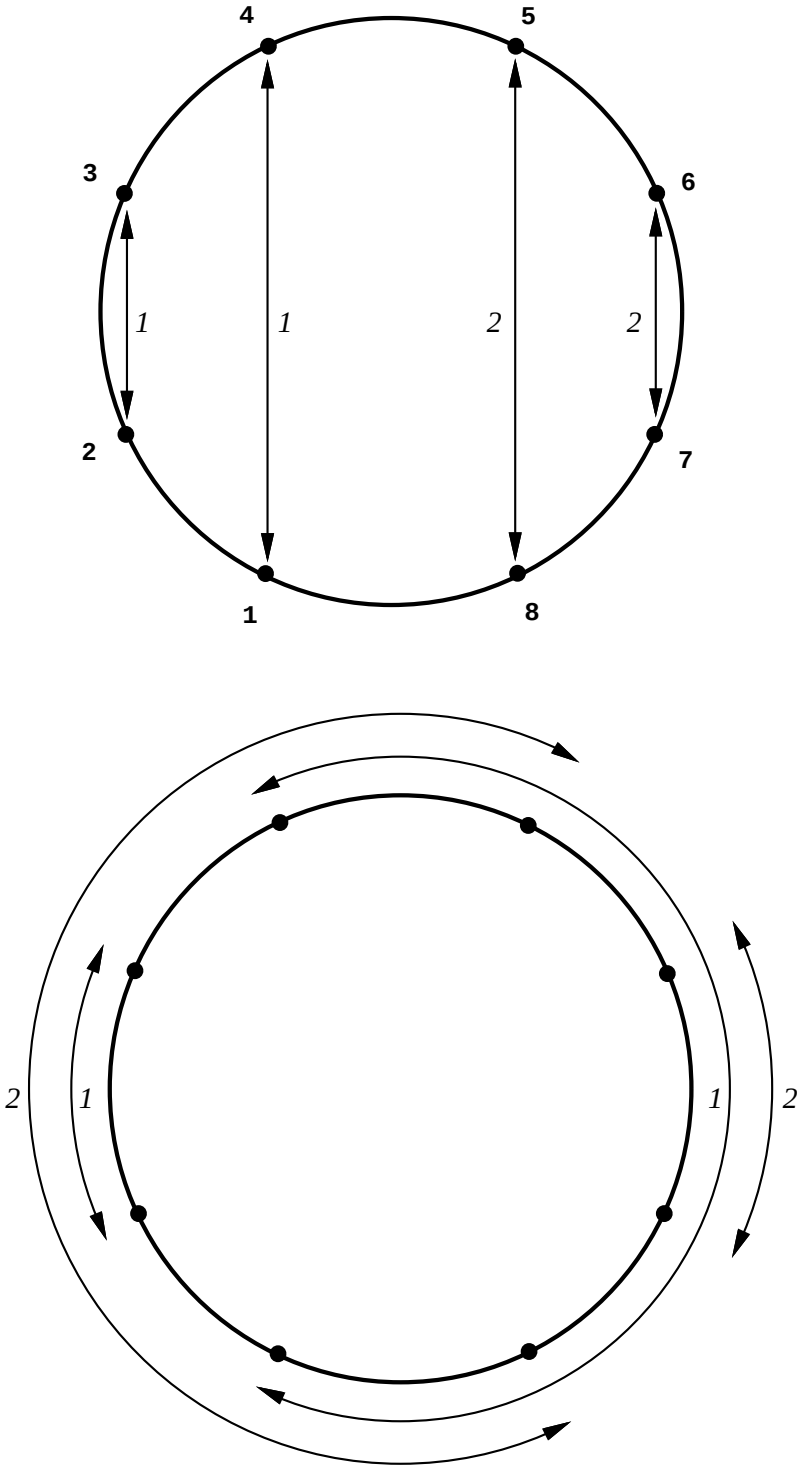


FIG. 1. *An instance of RING LOADING with optimal solution.*

PROPOSITION 4.2. *If $D_{g,h} \le C_g + C_h$ holds for each cut then there is a solution to* RELAXED RING LOADING *satisfying the capacity constraints.*

Proof. It will be useful in what follows to allow "cuts" of the form $\{g, g\}$, with capacity $2C_g$ and demand $D_{g,g} = 0$. The cut constraint for these cuts is thus equivalent to nonnegativity of the link capacities.

Assume the theorem fails and fix a counterexample with n minimal and, subject to the minimality of n, having the least possible number of nonzero demands.

Choose any nonzero demand—say, $d_{i,j}$—with $i < j$, and let $\{g, h\}$ minimize $M = D_{g,h} - C_g - C_h$ subject to $i \le g < h < j$; thus, $\{g, h\}$ is the tightest cut in the front route for $d_{i,j}$. (A cut $\{g, h\}$ is said to be "tight" if $D_{g,h} = C_g + C_h$.)

We propose to send $\min(d_{i,j}, M/2)$ of the demand $d_{i,j}$ around the front and, if $d_{i,j} > M/2$, send the remaining $d_{i,j} - M/2$ around the back. When the capacities have been decreased accordingly, we will have a new RELAXED RING LOADING instance with one less nonzero demand. If the new instance still satisfies the cut constraints, it will contradict minimality of the counterexample, proving the theorem.

Suppose that in the new instance some cut is violated. That cut must lie on the back route for $d_{i,j}$, since this demand has already been accounted for in cuts which it crosses, and cuts on the front route have sufficient slack by choice of M. Then we have a cut $\{g', h'\}$ with $[g', h') \cap [i, j) = \emptyset$ such that

$$D_{g',h'} + 2(d_{i,j} - M/2) > C_{g'} + C_{h'}$$

where all quantities are computed in the original instance.

Call the $\{g, h\}$ cut and the $\{g', h'\}$ cut "straight" and consider also the "diagonal" cuts $\{g, g'\}$ and $\{h, h'\}$. Every demand must cross at least as many of the two diagonal cuts as the two straight cuts, while $d_{i,j}$ crosses both diagonal cuts and neither straight cut. Hence,

$$\begin{aligned} & D_{g,g'} + D_{h,h'} \\ &\ge D_{g,h} + D_{g',h'} + 2d_{i,j} \\ &> C_g + C_h - 2(M/2) + C_{g'} + C_{h'} - 2(d_{i,j} - M/2) + 2d_{i,j} \\ &= C_g + C_{g'} + C_h + C_{h'} \end{aligned}$$

so that one of the diagonal cuts must have violated the cut constraint.

Note that nonviolation of cuts of the form $\{g, g\}$ assures us that the given routing of $d_{i,j}$ is actually possible, i.e., that no link capacity will become negative afterward. □

Given a set of demands, we now wish to find an assignment ϕ^* which minimizes L^* and satisfies the conclusion of Proposition 4.1. This can be done quickly by putting each link in a tight cut as follows.

First we compute the $\binom{n}{2}$ values $D_{g,h}$, $1 \le g < h \le n$; let the largest of these be M. Then $L^* \ge M/2$, but the ring with all capacities set to $M/2$ satisfies the cut constraint, so in fact $L^* = M/2$. We now take the links in any order (say, $\{1, 2\}$ through $\{n, 1\}$) and lower their capacities as much as possible; that is, define capacities $\{C_i\}$ recursively by

$$C_g = \max\left(\max h < g(D_{g,h} - C_h), \max h > g(D_{g,h} - M/2)\right),$$

noting that $C_g \ge 0$.

No realizable set $\{C_i'\}$ of capacities can have $C_i' \le C_i$ for every i and $C_j' < C_j$ for some j, since the least such j would be part of a bad cut. Hence any feasible assignment ϕ^* for these capacities is a minimal solution of the original RELAXED RING LOADING instance, and Proposition 4.1 applies. In particular, if $S = \{\{i,j\} : \ d_{i,j} \text{ is split by } \phi^*\}$, then every pair of chords in S crosses and, therefore, $|S| \le n/2$.

In fact, after reducing the capacities as above we can solve RELAXED RING LOADING to route each demand all front or all back until only mutually pairwise crossing demands remain. To see this, assume that there is still a parallel pair of unrouted demands and choose a link between them; fix a tight cut containing that link. At most one of the two parallel demands crosses the cut; the other can, and indeed must, be routed to miss the cut entirely.

In summary, our algorithm for solving RELAXED RING LOADING proceeds as follows:

1. Compute the $\binom{n}{2}$ values $D_{g,h}$, and $L^* = M/2$.
2. Compute minimal capacities $\{C_i\}$ as described above.
3. While there are pairs of parallel demands, find tightest cuts and route demands all front or all back, resetting capacities accordingly.
4. When only crossing cuts remain, route as much as possible by the front and the remainder by the back.

The running time of this procedure is approximately of order kn^2, where k is the number of nonzero demands; this is very fast for the parameter sizes that we require. See [10] for an even faster solution to problems akin to RELAXED RING LOADING.

In any case, our solution to RELAXED RING LOADING ends with at most only $n/2$ of the demands split. It therefore seems natural to compute ϕ^* and then "unsplit" the demands in S as gently as possible in order to get a near-optimal $\{0,1\}$ assignment for RING LOADING. This is exactly what we do.

5. Unsplitting. Henceforth ϕ^* will be a fixed, minimal solution to RELAXED RING LOADING with a set of split demands S as above. We seek a solution ϕ to RING LOADING which agrees with ϕ^* when $\phi^*(i,j) \in \{0,1\}$ and for which $L - L^*$ is as small as possible, where L is the ringload of ϕ.

If node i is not an endpoint of a split demand, then the difference between the loads on links $\{i-1,i\}$ and $\{i,i+1\}$ will not change as we pass from ϕ^* to ϕ. Hence, for the purpose of determining ϕ, we may as well delete vertex i and combine the two former links to form a single link whose load under the relaxed assignment is taken to be $\max(L^*_{i-1}, L^*_i)$. Proceeding in this fashion for each vertex not involved in a split demand, we are reduced to the case where n is even and $S = \{\{i,i+m\} : \ 1 \le i \le m\}$, with $m = n/2$.

Let us define u_i to be the amount of demand $d_{i,i+m}$ sent by ϕ^* via the front route, and v_i via the back, so that $u_i,\ v_i > 0$ and $u_i + v_i = d_{i,i+m}$. If ϕ routes $d_{i,i+m}$ by the front, then each link $\{j,j+1\}$ with $j \in [i,i+m)$ has its load incremented by v_i (the amount formerly sent around the back) relative to the relaxed assignment ϕ^*, while the rest of the link loads are decremented by v_i. Similarly, if demand $d_{i,i+m}$ is sent by the back route, the load of each link in $[i,i+m]$ is decreased by u_i while the rest are incremented by the same amount.

Hence if we set $z_i = v_i$ when $\phi(i,i+m) = 1$ and $z_i = -u_i$ otherwise, we have

$$L_j = L^*_j + \sum_{\substack{i \in [1,m] \\ j \in [i,i+m)}} z_i - \sum_{\substack{i \in [1,m] \\ j \in [i+m,i)}} z_i.$$

Notice that $L_j + L_{j+m} = L^*_j + L^*_{j+m}$ for all j. Thus $L \le 2L^*$ for *all* choices of ϕ, duplicating the performance ratio claimed by Cosares and Saniee [3], but we will do much better by choosing ϕ judiciously.

The optimal ϕ can be found by dynamic programming, but in practice we try every ϕ and choose the best one! There are at most $2^{n/2}$ choices for ϕ, a list easily exhausted for all currently contemplated SONET ring sizes. In effect, for our values of n (up to 32, possibly) the line between tractability and intractability lies not between polynomial and exponential but between exponential in n and exponential in n^2.

Our embarrassment, as theorists, is assuaged somewhat by the fact that there *is* a polynomial algorithm for finding an assignment ϕ which achieves the performance guaranteed by the following theorem.

THEOREM 5.1. *Let ϕ^* be a minimal solution, with ringload L^*, to the relaxed version of an instance of* RING LOADING. *Let D be the maximum magnitude of the demands split by ϕ^*. Then there is a $\{0,1\}$ assignment ϕ with ringload L which agrees with ϕ^*, except on split demands, and which satisfies $L - L^* \le \frac{3}{2}D$.*

Proof. We define z_i (hence ϕ) inductively, ensuring that $\sum_{i=1}^{k} z_i \in [-D/2, D/2]$ for all k, $1 \le k \le m$. This is always possible since, once $z_1, \ldots, z_{k-1}$ are defined and the partial sum $s = \sum_{i=1}^{k-1} z_i$ lies in the required interval, the two possible values of $\sum_{i=1}^{k} z_i$ lie on both sides of s and differ by only $u_k + v_k \le D$.

Put

$$M_k := \sum_{i=1}^{k} z_i - \sum_{i=k+1}^{m} z_i = 2\sum_{i=1}^{k} z_i - \sum_{i=1}^{m} z_i \in \left[-\frac{3}{2}D, \frac{3}{2}D\right]$$

and

$$M := \max_{1 \le k \le m} |M_k|,$$

then

$$L - L^* \le \max_j (L_j - L^*_j) = M \le \frac{3}{2}D. \qquad \square$$

The greedy unsplitting given in the proof of Theorem 5.1, when appended to our solution to RELAXED RING LOADING, gives the polynomial-time approximation algorithm which we call "Algorithm A."

Of course, the true optimum L^{opt} for the original RING LOADING problem is at least equal to L^*, so the theorem guarantees an additive error of at most a constant (3/2) times the maximum original demand irrespective of the value of n.

How good is this performance guarantee? This method can never achieve a multiplicative performance bound better than 2 relative to L^*, since the "square example" with $n = 4$, $d_{1,3} = d_{2,4} = 1$, and other demands 0 gives $L^* = 1$, $L^{\text{opt}} = 2$. Nor can we hope to get a factor better than 4/3 relative to L^{opt} due to the example shown in Fig. 1.

However, for larger n, if demands average $D/2$ in size then the typical demand adds $n/4 \cdot D/2$ to the total load when routed the short way; thus we expect the sum of the loads of all the links to be approximately $\binom{n}{2} \cdot n/4 \cdot D/2 \approx (D/16)n^3$, giving $L^* \ge (D/16)n^2$. Next to an optimum of order n^2, an additive error which does not depend on n at all looks pretty good; but we must again remember that n is never very large. For $n = 16$ this analysis allows a relative error of $(\frac{3}{2}D)/(16D) \approx 9\%$,

which is not so impressive. Of course this is pessimistic; the Cosares-Saniee algorithm allows 100% error in theory but does far better in practice. In any case, it would clearly be worth some effort to determine whether the constant 3/2 is best possible, and we tackle this problem in the last section.

First, however, we return to the $\{0,1\}$ demands case.

6. {0,1} Demands. In this section it will not complicate notation to allow many demands between two nodes of the ring, each of magnitude 1; we also allow capacities C_i for the links, not necessarily equal. A cut $\{g,h\}$ is said to be *even* if $C_g+C_h-D_{g,h} \equiv 0 \pmod 2$. In [6] feasibility is shown to be equivalent to the cut condition together with the following parity condition.

Parity condition. For every pair of links g,h, if g and h are each in a tight cut, then the cut $\{g,h\}$ is even.

THEOREM 6.1. *In the $\{0,1\}$ case, if we put $C_i = L$ for each link i, then* RING LOADING *is feasible with ringload $\leq L$ if and only if the cut and parity constraints are satisfied. If only the cut constraint is satisfied, then the optimal ringload is $L+1$. In any case, the algorithm A described above finds an optimal assignment.*

Proof. It is straightforward to verify that if a demand is assigned (all front or all back) without violating the cut condition, then the truth value of the parity condition is preserved. Since the parity condition is met when all demands are assigned, necessity is clear.

On the other hand, suppose that demands are assigned in accordance with Algorithm A until all remaining demands require splitting. Suppose there is at least one left, say, $d_{i,j}$; then there must be parallel cuts $\{g,h\}$ and $\{g',h'\}$ on each side of $d_{i,j}$ with $C_g + C_h - D_{g,h} = C_{g'} + C_{h'} - D_{g',h'} = 1$. Since the diagonal cuts must be tight, the parity condition is (twice) violated.

It remains only to observe that if the cut constraint is satisfied when $C_i = L$, then at $C_i = L+1$ we also satisfy the parity constraint since all the cuts have slack. □

7. The constant. Let β be the infimum of all reals α such that the following combinatorial statement holds: For all positive integers m and nonnegative reals $u_1,\ldots,u_m$ and $v_1,\ldots,v_m$ with $u_i + v_i \leq 1$, there exist $z_1,\ldots,z_m$ such that for every k, $z_k \in \{v_k, -u_k\}$ and

$$\left| \sum_{i=1}^{k} z_i - \sum_{i=k+1}^{m} z_i \right| \leq \alpha.$$

Then β is the "right" constant for Theorem 5.1, i.e., $L - L^* \leq \beta D$ for some choice of ϕ. Note that any choice of rational values for the u_i's and v_i's can actually occur (up to constant factor) from an instance of RING LOADING, since we can construct one as follows. Let M_j be the load actually incurred by link $\{j, j+1\}$ when the demands $d_{i,i+m} = u_i + v_i$ are split u_i front and v_i back. Let M be huge, and postulate additional "short" demands $d_{j,j+1} = M - M_j$ for each j, $1 \leq j \leq 2m$. Then any optimal RELAXED RING LOADING solution will send all the short demands by the one-link route; however, the sum of the link loads due to the other demands is constant since each has two routes of the same length. Thus splitting the other demands as given, so as to obtain the same load M on every link, is optimal, and it is easy to see that no other splitting can achieve uniform load.

We already know $\beta \le 3/2$ and the square example, where $m = 2$ and $u_1 = v_1 = u_2 = v_2 = 1/2$, shows that $\beta \ge 1$. (In fact, u_i's and v_i's chosen uniformly at random subject to the given constraints also force $\beta \ge 1$.)

The special case where $u_i = v_i$ for each i is interesting for several reasons. This means that ϕ^* is sending exactly half of each demand $d_{i,i+m}$ each way around the ring, giving us no clue how to unsplit them. Furthermore, this is the case which arises when (as in the square example) all of the nonzero demands in the original RING LOADING instance are mutually crossing.

The case $u_i = v_i$ thus gives rise to a new ring loading problem as well as the following new constant.

CROSSED RING LOADING

INSTANCE: Ring size $2m$ and nonnegative reals d_i, $1 \le i \le m$.
QUESTION: Find a map $\phi : \{1, 2, \ldots, m\} \longrightarrow \{0, 1\}$ which minimizes $L = \max_{1 \le k \le 2m} L_k$, where

$$L_k = \sum\{\phi(i)d_i : \ k \in [i, i+m)\} + \sum\{(1 - \phi(i))d_i : \ k \notin [i, i+m)\}.$$

Note that we have allowed real demands here (rationals would be fine, too) in order to handle nonintegral splits produced by a previous linear programming phase.

We define γ be the infimum of all reals α such that the following combinatorial statement holds: For all positive integers m and $x_1, \ldots, x_m \in [0, 1]$ there exist $y_1, \ldots, y_m$ such that for every k, $|y_k| = x_k$ and

$$\left| \sum_{i=1}^{k} y_i - \sum_{i=k+1}^{m} y_i \right| \le 2\alpha.$$

(Note that we have rescaled the combinatorial statement so that the x_i's lie in the unit interval instead of $[0, 1/2]$.)

We have $1 \le \gamma \le \beta \le 3/2$. For lack of a counterexample, the authors were moved to conjecture publicly that both constants are equal to 1. After an embarrassingly long interval we found a simple proof, given below, that $\gamma = 1$; thus we have the following theorem.

THEOREM 7.1. *Let K be the sum of the demands of an instance of* CROSSED RING LOADING. *Then there is an assignment ϕ (which can be found in time polynomial in m and the length of the demand descriptions) whose ringload L satisfies $L - K/2 \le D$.*

Proof. We must show that given $x_1, \ldots, x_m \in [0, 1]$ there are $y_1, \ldots, y_m$ such that for every k, $|y_k| = x_k$ and

$$\left| \sum_{i=1}^{k} y_i - \sum_{i=k+1}^{m} y_i \right| \le 2.$$

As in the asymmetric case, we can obtain a bound of 3 instead of 2 by greedy assignment; in this case that amounts to putting $y_k = x_k$ when $\sum_{i=1}^{k-1} y_i \le 0$ and $y_k = -x_k$ otherwise. We generalize this algorithm by choosing a real w instead of 0 as the "empty sum."

Specifically, for fixed $w \in [-1, 1]$, define y_k inductively by $y_k = x_k$ when $w + \sum_{i=1}^{k-1} y_i \le 0$ and $y_k = -x_k$ otherwise. Then $w + \sum_{i=1}^{k} y_i \in [-1, 1]$ for all k; let $f(w) := w + \sum_{i=1}^{m} y_i$.

Suppose that $f(w) = -w$; then

$$\begin{aligned}
&\sum_{i=1}^{k} y_i - \sum_{i=k+1}^{m} y_i \\
&= 2\sum_{i=1}^{k} y_i - \sum_{i=1}^{m} y_i \\
&= 2\left(w + \sum_{i=1}^{k} y_i\right) - \left(w + \sum_{i=1}^{m} y_i\right) - w \\
&\in [-2, 2]
\end{aligned}$$

as desired.

Since $f(-1) + (-1) \le 0 \le f(1) + 1$, the existence of a w for which $f(w) = -w$ would follow from the intermediate value theorem if f were continuous. Of course this is not the case; whenever a partial sum hits 0, some y_i's change sign and $f(w)$ may jump. (Since we have chosen y_i positive when the partial sum is 0, f will be continuous from the left.) However, it turns out that the *absolute value* of f is continuous.

Note first that when no partial sum is at 0, the derivative $f'(w)$ is 1. On the other hand suppose that $w = w_0$ is chosen such that one or more of the partial sums is zero; in particular, let $k \ge 0$ be minimal such that $w + \sum_{i=1}^{k} y_i = 0$. Then for sufficiently small ε, the signs of y_j and $w + \sum_{i=1}^{j} y_i$, for $j > k$, flip as we move from $w = w_0$ to $w = w_0 + \varepsilon$. Hence, taking $j = m$, we have that $\lim_{w \to w_0^+} f(w) = -f(w_0)$.

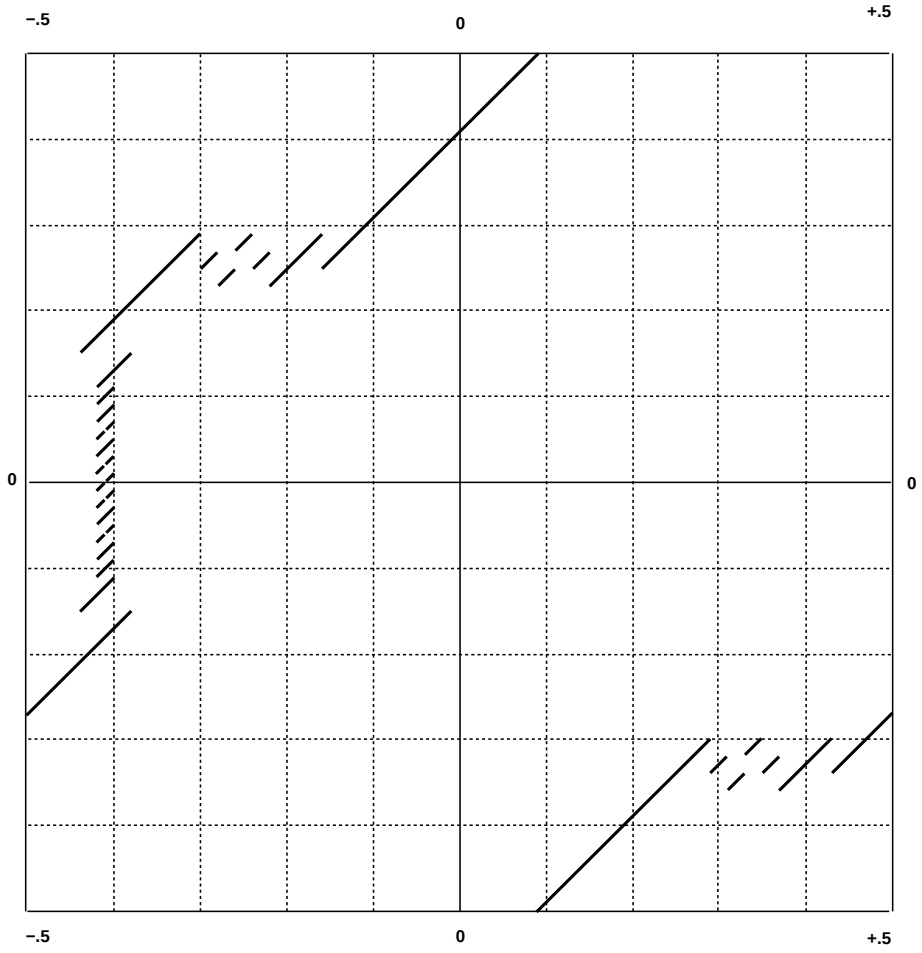
It follows that when any partial sum hits zero we will have $\lim_{w \to w_0^+} f(w) = -f(w_0)$; thus the function g given by $g(w) = |f(w)|$ will be continuous everywhere and differentiable except at finitely many points. The graph of g is a zig-zag, with derivative 1 where $g(w) = f(w)$ and -1 where $g(w) = -f(w)$.

Of course, if we define h by $h(w) = -w$, then the graph of h is a line of slope -1 from $(-1, 1)$ to $(1, -1)$ which must intersect the graph of g. Moreover, it must either intersect at a point where $g'(w) = 1$ or coincide with a segment of the graph of g of slope -1, in which case the leftmost point of the segment lies also on the graph of f. Either way we have a point w at which $-w = g(w) = f(w)$.

To complete the proof we need to demonstrate a fast algorithm for finding this w. To do this we set the y_i's one at a time while keeping a solution w in range. Specifically, at stage j we have values $y_1, \ldots, y_j$ fixed and $a_j \le w \le b_j$, with $g(a_j) \le -a_j$ and $g(b_j) \ge -b_j$; of course this holds at stage 0 with $a_0 = -1$, $b_0 = 1$. At stage $j + 1$, if $a_i + \sum_{i=1}^{j} y_i$ and $b_i + \sum_{i=1}^{j} y_i$ are both positive, then perforce we set $y_{j+1} = -x_{j+1}$; if both are ≤ 0, then we put $y_{j+1} = x_{j+1}$. In either of these cases we set $a_{j+1} = a_j$ and $b_{j+1} = b_j$.

Otherwise $s := -\sum_{i=1}^{j} y_i$ lies in the half-open interval $[a_j, b_j)$. If $g(s) < -s$, we put $y_{j+1} = -x_{j+1}$ and set $a_{j+1} = s$ and $b_{j+1} = b_j$; if $g(s) \ge -s$, put $y_{j+1} = x_{j+1}$ and set $a_{j+1} = a_j$ and $b_{j+1} = s$. In any case the inductive conditions are preserved, the intervals $[a_j, b_j]$ are nested downward, and at stage m all the y_i's are correctly set. □

We have shown $\gamma = 1$, but the proof above will not work for β, as $g = |f|$ is no longer continuous in the asymmetric case. Even so, we may gain by replacing f by a multivalued function F, defined by $z \in F(w)$ if $z = w + \sum_{i=1}^{n} z_i$ for any $z_1, \ldots, z_n$ which keep the sums $w + \sum_{i=1}^{k} z_i$ within bounds.



$i=$	1	2	3	4	5	6	7	8	9	10	11	12	13
u_i	.17	.71	.25	.71	.23	.75	.76	.21	.75	.21	.73	.25	.50
v_i	.83	.23	.75	.21	.77	.21	.24	.73	.25	.71	.27	.71	.50

FIG. 2. *An example in which additive error of D cannot quite be achieved.*

Then the graph of F will be a union of slope-1 line segments, each corresponding to an assignment of z_i's. The sum of the lengths of these segments will be at least $\sqrt{2}$ since $F(w)$ always takes on at least one value, and in practice—and in virtually any random model—the segments will practically always intersect the line from $(-.5, .5)$ to $(.5, -.5)$ at least once, providing a solution to RING LOADING which is within D of L^*.

However, it is just barely possible to choose values u_i and v_i for which the diagonal line sneaks through between the line segments of the graph of F. A set of such values, for $m = 13$, is given in Fig. 2 along with the corresponding graph of F. On the graph, each of the 2^{13} routings is represented by a diagonal line segment, often null,

TABLE 1

n	k	C%	C-B	B=C	A-C	A=C	A-B	A=B	Bt	At	Ct
8	28	100%	.0054	63%	.0110	19.4%	.0160	10.7%	.0001	.0002	.002
12	66	99%	.0013	85%	.0036	21.2%	.0051	19.5%	.0004	.0005	.1
16	120	96%	.0003	94%	.0017	22.3%	.0023	20.7%	.0016	.0016	.45
20	190	93%	.00014	96%	.0010	26.2%	.0015	23.6%	.0036	.0038	.78
24	276	93%	.00002	99%	.0007	27.2%	.0008	24.8%	.007	.007	.84
28	378	92%	.00000	99%	.0004	28.3%	.00056	25.5%	.013	.012	.92
32	496	92%	.00000	99%	.0002	29.2%	.00037	26.4%	.02	.019	1.1

indicating the final sum $w + \sum_{i=1}^{m} z_i$ as a function of w, for just those values of w for which all partial sums $w + \sum_{i=1}^{k} z_i$ lie between $-.5$ and $.5$.

With this general definition of the multifunction F, a crossing of the diagonal is necessary as well as sufficient to get a solution within 2 of L^*. Hence the example shows that β is at least 1.01. This lower bound can certainly be raised somewhat but it is far from clear that the true value of β is anywhere near $3/2$.

8. Conclusions. Experimental results show that indeed our proposed algorithm is adequately fast and, when applied to random examples small enough to compute L^{opt}, produces a ringload very close to optimal. We have never managed to produce a random example with $L > L^* + D$ even though our theorem guarantees only $L \leq L^* + \frac{3}{2}D$, and we doubt such an instance will ever be seen in practice.

Hence, even though the mathematics refuses to cooperate, *we* guarantee $L \leq L^* + D$.

Table 1 above exhibits the results of testing our algorithm, which we call "Algorithm A," on uniformly random data. Alongside "A" we ran a linear programming algorithm, "Algorithm B," in order to compute the lower bound given by the RELAXED RING LOADING solution. To find the optimum ringload and for purposes of comparison, we also tested "Algorithm C," which recursively looks for an optimal solution. In most cases Algorithm C was not enormously slower than A, but it became hopelessly stuck in some cases, leaving us with no value for the optimal ringload.

For each set of parameters, 1000 cases were run. The interpretation of the columns of the table is as follows:

n: number of nodes in the ring,
k: number of demands,
C%: percentage of runs in which the optimum was found,
C-B: average error of LP bound relative to optimum,
B=C: percentage of runs in which LP bound = optimum,
A-C: average error of our algorithm relative to optimum,
A=C: percentage of cases in which A hit the optimum,
A-B: average error of LP bound relative to A,
A=B: percentage of cases in which A achieves LP bound,
Bt: average running time for the LP algorithm,
At: average running time for Algorithm A,
Ct: average running time for Algorithm C.

The fourth through seventh columns are computed only for those rounds in which the optimum was found; that creates a bias, especially for the column labelled B=C, since we will probably never get equality when Algorithm C fails. The run time for Algorithm C includes cases where it failed to find the optimum, and it was terminated after 10 seconds of CPU time on any one run.

Acknowledgments. The authors have had the benefit of valuable conversations with Noga Alon and Milena Mihail.

REFERENCES

[1] J. Babcock, *SONET: A practical perspective*, Business Comm. Rev., Sept. (1990), pp. 59–63.
[2] S. Cosares, I. Saniee, and O. Wasem, *Network planning with the SONET toolkit*, Bellcore EXCHANGE, Sept./Oct. (1992), pp. 8–13.
[3] S. Cosares and I. Saniee, *An optimization problem related to balancing loads on SONET rings*, Telecommunications Systems, 3 (1994), pp. 165–181.
[4] A. Frank, *Connectivity and network flows*, in Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 111–177.
[5] A. Frank, *Edge-disjoint paths in planar graphs*, J. Combin. Theory Ser. B, 38 (1985), pp. 164–178.
[6] A. Frank, T. Nishizeki, N. Saito, H. Suzuki, and E. Tardos, *Algorithms for routing around a rectangle*, Discrete Appl. Math., 40 (1992), pp. 363–378.
[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.
[8] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194.
[9] H. Okamura and P. D. Seymour, *Multicommodity flows in planar graphs*, J. Combin. Theory Ser. B, 31 (1981), pp. 75–81.
[10] H. Ripphausen-Lipa, D. Wagner, and K. Weihe, *Efficient algorithms for disjoint paths in planar graphs*, in Combinatorial Optimization, DIMACS Series in Discrete Math. Theoret. Comput. Sci. 20, W. Cook, L. Lovász, and P. Seymour, eds., AMS, Providence, RI, 1995, pp. 295–354.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 15–40, February 1998

002

INTERFERENCE-MINIMIZING COLORINGS OF REGULAR GRAPHS*

P. C. FISHBURN†, J. H. KIM†, J. C. LAGARIAS†, AND P. E. WRIGHT†

Abstract. Communications problems that involve frequency interference, such as the channel assignment problem in the design of cellular telephone networks, can be cast as graph coloring problems in which the frequencies (colors) assigned to an edge's vertices interfere if they are too similar. The paper considers situations modeled by vertex-coloring d-regular graphs with n vertices using a color set $\{1, 2, \ldots, n\}$, where colors i and j are said to *interfere* if their circular distance $\min\{|i-j|, n-|i-j|\}$ does not exceed a given threshold value α. Given a d-regular graph G and threshold α, an interference-minimizing coloring is a coloring of vertices that minimizes the number of edges that interfere. Let $I_\alpha(G)$ denote the minimum number of interfering edges in such a coloring of G. For most triples (n, α, d), we determine the minimum value of $I_\alpha(G)$ over all d-regular graphs and find graphs that attain it. In determining when this minimum value is 0, we prove that for $r \geq 3$ there exists a d-regular graph G on n vertices that is r-colorable whenever $d \leq (1 - \frac{1}{r})n - 1$ and nd is even. We also study the maximum value of $I_\alpha(G)$ over all d-regular graphs and find graphs that attain this maximum in many cases.

Key words. graph coloring, interference threshold, regular graph

AMS subject classifications. 05B99, 05C35

PII. S089548019427545X

1. Introduction.

This paper is motivated by telecommunication problems such as the design of planar regions for cellular telephone networks and the assignment of allowable frequencies to the regions. In our graph abstraction, vertices are regions, edges are pairs of contiguous regions, and colors correspond to frequencies. We presume that every region has the same number d of neighbors, which leads to considering degree-regular graphs. Interference occurs between two regions if they are neighbors and their frequencies lie within an interference threshold. We adopt the simplifying assumption that the number of colors available equals the number n of regions, and let α denote the threshold parameter so that colors i and j in $\{1, 2, \ldots, n\}$ interfere precisely when their circularly measured scalar distance is less than or equal to α. Precedents for the use of circularly measured distance in graph coloring include Vince [20] and Guichard and Krussel [11].

Our formulation leads to several interesting graph-theoretic problems. One is to determine for any given d-regular graph G and threshold α the minimum number $I_\alpha(G)$ of interfering edges over the possible colorings of G. Another is: given parameters n, α, and d, determine the minimum and maximum values of $I_\alpha(G)$ and find graphs G that attain these values. We focus on the latter problem. More specifically, let $\mathcal{G}(n, d)$ denote the set of undirected d-regular graphs on n vertices, which have no loops or multiple edges, but may be disconnected. We wish to determine the (global) *minimum interference level* $\ell(n, \alpha, d)$, which is the minimum of $I_\alpha(G)$ over $\mathcal{G}(n, d)$. For comparison purposes, we also wish to determine the (global) *minimax interference level* $L(n, \alpha, d)$, which is the maximum of $I_\alpha(G)$ over $\mathcal{G}(n, d)$. This latter problem

*Received by the editors October 7, 1994; accepted for publication (in revised form) January 6, 1997.

http://www.siam.org/journals/sidma/11-1/27545.html

†AT&T Labs-Research, 180 Park Avenue, Florham Park, NJ 07932 (fish@research.att.com, jhk@research.att.com, jcl@research.att.com, pew@research.att.com).

measures how badly off you would be if an adversary gets to choose $G \in \mathcal{G}(n,d)$, and you can then color G to minimize interference.

Our graph-theoretic model is an approximation to the frequency assignment problem for cellular networks studied in Benveniste et al. [1]. In that paper the network of cellular nodes is viewed as vertices of a hexagonal lattice Λ in $\mathbb{R}^2$, and the graph G is specified by a choice of sublattice Λ' of Λ, with $n = |V(G)|$ being the index of the sublattice Λ' in Λ. More precisely, the vertices of G are cosets of Λ/Λ', and we draw an edge between two cosets if the cosets are "close" in the sense that they contain vectors $\mathbf{v}$, $\mathbf{v}'$, respectively, with $||\mathbf{v} - \mathbf{v}'|| < x$, where $||\cdot||$ is a given norm on $\mathbb{R}^2$ and x is a cutoff value. Such graphs[1] G are d-regular for some value of d; the usual nearest-neighbors case gives $d = 6$: see Bernstein, Sloane, and Wright [2]. The frequency spectrum is also divided into cosets (modulo n), and nodes in the same coset (mod Λ') are assigned a fixed coset of frequencies (mod n). In cellular problems the graph G is fixed (depending on Λ'). Typical parameters under consideration are $10 \le n \le 30$, $d = 6$, and n/α about 2 or 3. From this standpoint the quantities $\ell(n,\alpha,d)$ and $L(n,\alpha,d)$ represent lower and upper bounds for attainable levels of interference.

Related coloring problems motivated by the channel assignment problem are studied in Hale [12], Cozzens and Roberts [6], Bonias [4], Liu [14], Tesman [17], Griggs and Liu [9], Raychaudhuri [15], Troxell [18], and Guichard [10] among others. Roberts [16] surveys the earlier part of this work. Factors that distinguish prior work from the present investigation include our focus on regular graphs and the inevitability of interference when certain relationships hold among n, α and d.

Our main results give near-optimal bounds for $\ell(n,\alpha,d)$ and $L(n,\alpha,d)$ and identify d-regular graphs and colorings that attain extremal values. Many interference-minimizing designs use only a fraction of the available colors or frequencies. The most common number of colors used in these optimal designs is

$$\gamma = \left\lfloor \frac{n}{\alpha+1} \right\rfloor ,$$

which is the maximum number of mutually noninterfering colors from $\{1,2,\ldots,n\}$ at threshold α. Detailed statements of theorems for $\ell(n,\alpha,d)$ and $L(n,\alpha,d)$ appear in section 2. Proofs follow in sections 3 to 7.

In the course of our analysis we derive a graph-theoretic result of interest in its own right, which is a condition for the existence of a d-regular graph having chromatic number $\le r$.

THEOREM 1.1. *If $r \ge 3$, then $\mathcal{G}(n,d)$ contains an r-colorable graph if nd is even and*

$$d \le \begin{cases} (1-\frac{1}{r})n - 1 & \text{if } r \text{ divides } n+1 \ , \\ (1-\frac{1}{r})n & \text{otherwise} \ . \end{cases}$$

This result is proved in section 5, and the proof can be read independently of the rest of the paper. Note that if nd is odd then $\mathcal{G}(n,d)$ is the empty set.

We preface the results in the next section with a few comments to indicate where we are headed. The case $\alpha = 0$ corresponds to no interference because the number of

[1] The graph G represents a fundamental domain of Λ/Λ'. In the cellular terminology, a fundamental domain for Λ/Λ' is called a "reuse group." More generally, a "reuse group" is a collection of contiguous cells that exhausts all frequencies, with no two cells in the group using the same frequency.

available colors equals the number of vertices, and therefore $\ell(n,0,d) = L(n,0,d) = 0$. We assume that $\alpha \geq 1$ in the rest of the paper.

For degrees near 0 or n, namely $d = 0, 1, n-2$ or $n-1$, the set $\mathcal{G}(n,d)$ contains only one unlabeled graph, so these cases are essentially trivial. We note at the end of section 4 that

$$\ell(n,\alpha,n-1) = L(n,\alpha,n-1) = \lfloor \frac{n}{\gamma} \rfloor \left(n - \frac{1}{2}\gamma \left(\lfloor \frac{n}{\gamma} \rfloor + 1 \right) \right) \ . \tag{1.1}$$

Our first main result in the next section, Theorem 2.1, applies to degree 2 and shows that most values of ℓ and L for $d = 2$ equal 0. A notable exception is that $L(n,2,2)$ is approximately $n/3$.

Subsequent results focus on $d \geq 3$, where we use the maximum number of noninterfering colors γ to express the results. The case $\gamma = 1$ is trivial because then all colors interfere with each other, so that $\ell = L = \#(\text{edges of } G) = nd/2$. For $\gamma \geq 2$, $\ell(n,\alpha,d)$ for most values of (n,α,d) is approximately

$$\max\left(\frac{nd}{2} - \frac{n^2}{2}\left(\frac{\gamma - 1}{\gamma} \right),\ 0 \right) \ .$$

Moreover, $L(n,\alpha,d) = 0$ whenever $\gamma > d$, whereas if n is much larger than d, and d is somewhat larger than γ, then $L(n,\alpha,d)$ is approximately $nd/(2\gamma)$.

Extremal graphs which attain $\ell(n,\alpha,d)$ when $\ell > 0$ are usually connected, and the associated coloring can often be achieved using γ noninterfering colors. On the other hand, graphs that attain $L(n,\alpha,d)$ when $L > 0$ are usually disconnected and contain many copies of the complete graph K_{d+1}. There are exceptions, however.

Our results imply that there is often a sizable gap between the values of ℓ and L. The smallest instance of $\ell < L$ occurs at $(n,\alpha,d) = (6,2,2)$ where $\ell = 0$ and $L = 2$. Figure 1.1 shows the two graphs in $\mathcal{G}(6,2)$ with interference-minimizing colorings for $\alpha = 2$.

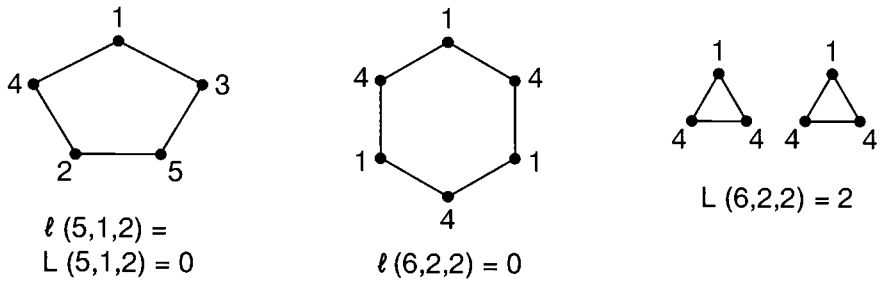


FIG. 1.1.

A qualitative comparison of the regions where ℓ and L equal 0 and are positive is given in Figure 1.2, where the coordinates are d/n and γ/n.

2. Main results. An undirected graph is *simple* if it has no loops or multiple edges. Let $\mathcal{G}(n,d)$ denote the set of d-regular graphs on n vertices which are simple but which are not necessarily connected. Let $[n] = \{1, 2, \ldots, n\}$ be a set of n colors with *circular distance measure*

$$D(i,j) = \min\{|i-j|, n - |i-j|\} \ ,$$

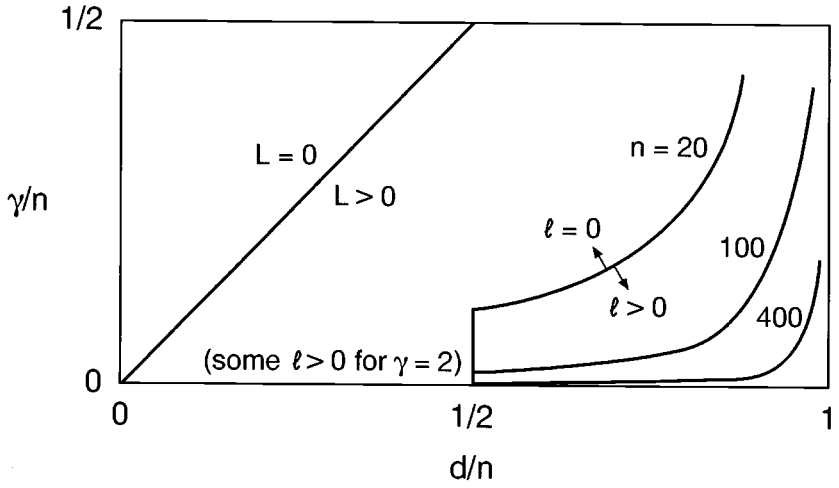


FIG. 1.2. *Zero and positive regions.*

and let $\alpha \in \{0, 1, \ldots\}$ be the threshold-of-interference parameter. A *coloring* of the vertex set $V(G)$ of graph $G = (V(G), E(G))$ in $\mathcal{G}(n, d)$ is a map $f : V(G) \to [n]$. The *interference* $I_\alpha(G, f)$ *of coloring* f *of* G *at threshold* α is

$$I_\alpha(G, f) := |\{\{x, y\} \in E(G) : D(f(x), f(y)) \le \alpha\}| \ .$$

The minimum interference in G at threshold α is

$$I_\alpha(G) := \min_{f:V(G)\to[n]} I_\alpha(G, f) \ .$$

We study the (global) *minimum interference level*

$$\ell(n, \alpha, d) := \min_{G\in\mathcal{G}(n,d)} I_\alpha(G), \tag{2.1}$$

and the (global) *minimax interference level*

$$L(n, \alpha, d) := \max_{G\in\mathcal{G}(n,d)} I_\alpha(G) \ . \tag{2.2}$$

We first note restrictions on the parameter space. Since all graphs in $\mathcal{G}(n, d)$ have $nd/2$ edges, it follows that

$$n \text{ and } d \text{ cannot both be odd} \ . \tag{2.3}$$

We restrict attention to the threshold range

$$1 \le \alpha \le \frac{n}{2} - 1 \ , \tag{2.4}$$

because $\alpha \ge n/2$ implies that all colors interfere. Thus

$$\gamma := \lfloor \frac{n}{\alpha + 1} \rfloor \ge 2 \ . \tag{2.5}$$

Our first result concerns ℓ and L for degree 2.

THEOREM 2.1. *Let $d = 2$.*

(a) *For all $\gamma \geq 2$,*

$$\ell(n, \alpha, 2) = 0 \ . \tag{2.6}$$

(b) *For all $\gamma \geq 3$,*

$$L(n, \alpha, 2) = 0 \ . \tag{2.7}$$

(c) *If $\gamma = 2$, and $n = 3M + j$ with $0 \leq j \leq 2$, then*

$$L(n, \alpha, 2) = \begin{cases} M & \text{if } j = 0, \text{ or } j = 2 \text{ with} \\ & \alpha \geq (2n-4)/5, \\ M-1 & \text{if } j = 1, \text{ or } j = 2 \text{ with} \\ & \alpha < (2n-4)/5 \ . \end{cases} \tag{2.8}$$

This is proved in section 3.

We now consider d in the range

$$3 \leq d \leq n-3$$

for the minimum interference level ℓ. The cases of $\gamma = 2$ and $\gamma \geq 3$ are treated separately. We obtain an almost complete answer for $\gamma = 2$.

THEOREM 2.2. *Suppose that $\gamma = 2$.*

(a) *If n is even, then*

$$\ell(n, \alpha, d) = 0 \quad \text{if} \quad d \leq \frac{n}{2} \ ,$$

and

$$\ell(n, \alpha, d) = \begin{cases} \frac{nd}{2} - \frac{n^2}{4} & \text{if } d > \frac{n}{2} \text{ and} \\ & \frac{n}{2} \text{ is even, or } \frac{n}{2} \text{ and } d \text{ are both odd,} \\ \frac{nd}{2} - \frac{n^2}{4} + 1 & \text{if } d > \frac{n}{2} \text{ and } \frac{n}{2} \text{ is odd} \\ & \text{and } d \text{ is even} \ . \end{cases} \tag{2.9}$$

(b) *If n is odd, then*

$$\ell(n, \alpha, d) = 0 \ \ \text{if} \ \ d < n - 2\alpha \ ,$$

and

$$\ell(n, \alpha, d) = \frac{nd}{2} - \frac{n^2}{4} + \frac{1}{4} \ \ \text{if} \ \ d > \frac{n}{2} \ . \tag{2.10}$$

(c) *If n is odd and in the remaining range $n - 2\alpha \leq d \leq \frac{n}{2}$, then $\ell(n, \alpha, d) \leq \frac{d}{2}$. Furthermore,*

(i) *$\ell(n, \alpha, d) = 0$ if there is an integer $2s + 1 \geq 5$ such that*

$$\alpha \leq \left(\frac{s}{2s+1}\right) n - 1 \ \ \text{and} \ \ d \leq \left(\frac{2}{2s+1}\right) n \ ;$$

(ii) $\ell(n,\alpha,d) \le \frac{d}{2} - 1$ *if* $d \ge 8$, $\frac{d}{2}$ *is even, and there is an integer* $4s+1 \ge 5$ *such that*

$$\alpha \le \left(\frac{2s}{4s+1}\right) n - 1 \quad \textit{and} \quad d = \left(\frac{2}{4s+1}\right)(n+1) \ ;$$

(iii) $\ell(n,\alpha,d) = \frac{d}{2}$ *for* $\alpha = (n-3)/2$.

Case (c) above is the only case not completely settled. Instances of it are illustrated in Figure 2.1. The number beside each vertex clump gives the color assigned to those vertices, and the number on a line between noninterfering clumps is the number of edges between them. Case analyses, omitted here, show that no improvements are possible in part (c) of the theorem when $n \le 21$. Given $n \le 21$, (i) has three realizations, namely $\ell(15,5,6) = \ell(21,7,8) = \ell(21,8,6) = 0$, (ii) has only the realization at the bottom of Figure 2.1, and $\ell = d/2$ for all other cases.

We remark that the bounds on $\ell(n,\alpha,d)$ for $d > n/2$ are obtained using a variant of Turán's theorem on extremal graphs (Turán, [19]; Bondy and Murty, [3, p. 110]). Theorem 2.2 is proved in section 4.

We now consider the minimum interference level ℓ when $\gamma \ge 3$. To handle this case we use Theorem 1.1, which is proved in section 5. Let p and q be the unique nonnegative integers that satisfy

$$n = p\gamma + q \quad \text{with} \ \ 0 \le q < \gamma \ ,$$

that is,

$$p = \lfloor \frac{n}{\gamma} \rfloor \ \text{ and } \ q = n - \gamma \lfloor \frac{n}{\gamma} \rfloor \ . \tag{2.11}$$

Our bounds for $\gamma \ge 3$ are given in the next two theorems for $q = 0$ and $q > 0$, respectively, and are proved in section 6. The $q = 0$ case is somewhat simpler.

THEOREM 2.3. *Suppose that* $\gamma \ge 3$ *and that* γ *divides* n, *i.e.,* $q = 0$.

(a) *If* $d \le n - p$, *then*

$$\ell(n,\alpha,d) = 0 \ .$$

(b) *If* $d > n - p$, *then*

$$\ell(n,\alpha,d) = \begin{cases} \frac{n(d-n+p)}{2} & \textit{if } n-d \textit{ is odd or if} \\ & n-d \textit{ is even and } p \textit{ is even}, \\ \frac{n(d-n+p)}{2} + \frac{\gamma}{2} & \textit{if } n-d \textit{ is even and } p \textit{ is odd} \ . \end{cases}$$

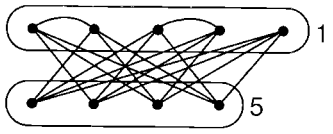
THEOREM 2.4. *Suppose that* $\gamma \ge 3$ *and* γ *doesn't divide* n, *i.e.,* $q \ge 1$.

(a) *If* $d < n - p$, *then*

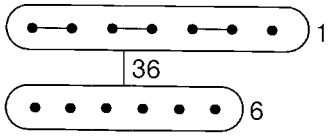
$$\ell(n,\alpha,d) = \begin{cases} 0 & \textit{if } d < n-p-1, \textit{ or } d = n-p-1 \\ & \textit{and } q < \gamma - 1, \\ \frac{p}{2} & \textit{if } d = n-p-1 \textit{ and } q = \gamma - 1 \ . \end{cases}$$

(b) *If* $d \ge n - p$, *then*

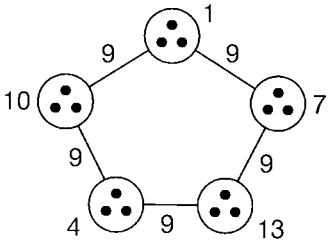
$$\ell(n,\alpha,d) = \frac{n(d-n+p)}{2} + \theta,$$



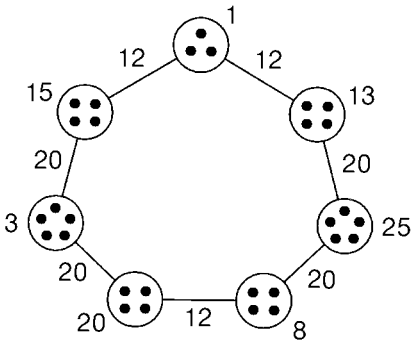
ℓ (9,3,4) = d/2 = 2: augmented bipartite
α = (n – 3)/2



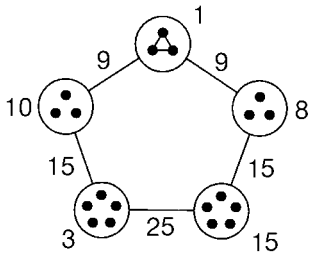
ℓ (13,4,6) = d/2 = 3: augmented bipartite
α = (n – 5)/2



(i) ℓ (15,5,6) = 0
s = 2



(i) ℓ (29,11,8) = 0
s = 3



(ii) ℓ (19,6,8) = d/2 – 1 = 3
s = 1
d = 8
n = (4s + 1) d/2 – 1 = 19

FIG. 2.1. $\gamma = 2$, *n odd*, $n - 2\alpha < d < n/2$.

where

$$\theta = \begin{cases} \frac{q(p+1)}{2} & if\ n-d\ is\ odd, \\ \frac{q(p+2)}{2} & if\ n-d\ is\ even\ and\ p\ is\ even, \\ \frac{pq+\gamma}{2} & if\ n-d\ is\ even\ and\ p\ is\ odd\ . \end{cases} \tag{2.12}$$

We turn next to results for the minimax interference level L. We first distinguish cases where $L = 0$ from cases where $L > 0$.

THEOREM 2.5. *Suppose that* $3 \le d \le n-2$. *Then*

(a) $L(n,\alpha,d) = 0$ *whenever* $\gamma > d$ *and also when* $\gamma = d$ *and* $n < 2(d+1)$;

(b) $L(n,\alpha,d) > 0$ *for* $\gamma \le d$ *whenever* $n \ge 2(\gamma+1)$.

The only cases in the parameter range $1 \le \alpha \le \frac{n}{2} - 1$ and $\gamma \ge 2$ not settled by this theorem are those with

$$\gamma = d - a \quad \text{and} \quad n = 2(d-a) \text{ or } 2(d-a)+1, \text{ where } \quad a > 0 \ . \tag{2.13}$$

Both $L = 0$ and $L > 0$ occur in this exceptional case, e.g., for $a = 1$, $L(8,1,5) = 0$ while $L(7,1,4) = 1$.

Our final main result provides bounds for L. Set

$$Q = d + 1 - \gamma \lfloor \frac{d+1}{\gamma} \rfloor,$$

and

$$W = n - (d+1)\lfloor \frac{n}{d+1} \rfloor \ .$$

In view of Theorem 2.5 we consider only the range that $2 \le \gamma \le d$.

THEOREM 2.6. *Suppose that* $3 \le d \le n-1$ *and that* $2 \le \gamma \le d$. *Then*

$$L(n,\alpha,d) \ge \frac{1}{2\gamma} \lfloor \frac{n}{d+1} \rfloor ((d+1)(d+1-\gamma) + Q(\gamma - Q)) - \frac{1}{2} W(d+1-W) \ ,$$

and

$$L(n,\alpha,d) \le \frac{1}{2\gamma} \left(\frac{d}{n-1} \right) (n(n-\gamma) + q(\gamma - q)) \ .$$

In the special case that $d+1$ divides n, these bounds can be written more simply as

$$\frac{nd}{2\gamma} - \frac{n(\gamma-1)}{2\gamma} + \frac{nQ(\gamma-Q)}{2\gamma(d+1)} \le L(n,\alpha,d) \le \frac{nd}{2\gamma} - \frac{nd(\gamma-1)}{2\gamma(n-1)} + \frac{dq(\gamma-q)}{2\gamma(n-1)} \ .$$

This applies in particular when $d = n-1$, in which case the upper and lower bounds coincide, yielding (1.1). If n is substantially larger than d, and d is somewhat larger than γ, then L is closely approximated by $nd/2\gamma$.

Theorems 2.5 and 2.6 are proved in section 7.

3. Elementary facts: Theorem 2.1. We derive general conditions that guarantee $\ell = 0$ or $L = 0$, and then analyze degree-2 graphs (Theorem 2.1).

LEMMA 3.1. *If* $1 \le \alpha < \frac{n}{2}$, *then*

$$\text{(a)} \quad \ell(n,\alpha,d) = 0 \quad \textit{whenever} \quad d < n - 2\alpha \ , \tag{3.1}$$

and

$$\text{(b)} \quad \ell(n,\alpha,d) = 0 \quad \textit{whenever} \quad d \le \frac{n}{2} \textit{ and } n \textit{ is even.} \tag{3.2}$$

Proof. (a) Given $d < n - 2\alpha$, let $V(G) = \{1, 2, \ldots, n\}$ and consider the coloring $f(i) = i$ for every i. We construct a suitable G starting with the edge set

$$E = \{\{i, j\} : i, j \in [n],\ i \neq j,\ \text{with } D(i, j) \geq (n + 1 - d)/2\}\ .$$

If n is odd, or if n is even and d is odd, let $E(G) = E$. Then every vertex has degree d and every edge has $D > \alpha$, so $\ell(n, \alpha, d) = 0$. If n and d are both even, so $\alpha \leq (n - d)/2 - 1$, let $E(G) = (E \cup \{\{i, j\} : D(i, j) = (n - d)/2\}) \backslash \{\{1, (n/2) + 1\}, \{2, (n/2) + 2\}, \ldots, \{(n/2), n\}\}$. Again, every vertex has degree d and every edge has $D > \alpha$, so $\ell(n, \alpha, d) = 0$.

(b) Let χ_G denote the chromatic number of the graph G. The definition implies that

$$\ell(n, \alpha, d) = 0 \ \text{ if } \ \chi_G \leq \gamma \text{ for some } G \in \mathcal{G}(n, d)\ . \tag{3.3}$$

If n is even and $d \leq (n/2)$, then $\mathcal{G}(n, d)$ contains a bipartite graph with $n/2$ vertices in each part, so $\chi_G = 2$, and (b) follows from (3.3), since $\gamma \geq 2$. □

We remark that the construction in part (a) uses all n colors, and when $d \geq n - 2\alpha$ this same construction gives many interfering edges. It is natural to consider the opposite extreme, which is to use only a maximal set of $\gamma = \lfloor n/(\alpha + 1) \rfloor$ noninterfering colors. This leads to part (b).

The restriction in part (b) that n be even is crucial, because no d-regular bipartite graph exists for odd n. Indeed, there are exceptions where $\ell(n, \alpha, d) > 0$ for some $d < n/2$ with n odd (see Theorems 2.1 and 2.2). These exceptions occur when $\gamma = 2$, but are not an issue for $\gamma \geq 3$.

We obtain bounds on the minimax interference level L using the following well-known bound for the chromatic number χ_G of a graph G.

PROPOSITION 3.2. *For every finite simple graph G,*

$$\chi_G \leq \Delta_G + 1\ , \tag{3.4}$$

where Δ_G is the maximum degree of a vertex of G. Furthermore, $\chi_G \leq \Delta_G$ provided that no connected component of G is an odd cycle or a complete graph.

Proof. For the proof, see Brooks [5] and Bondy and Murty [3, pp. 118 and 122]. □

This result immediately yields the following condition for the minimax interference level $L = 0$.

LEMMA 3.3. *If $1 \leq \alpha < (n/2)$, then*

$$L(n, \alpha, d) = 0 \text{ whenever } \gamma > d\ . \tag{3.5}$$

Proof. The definition of $L(n, \alpha, d)$ gives

$$L(n, \alpha, d) = 0 \text{ if } \chi_G \leq \gamma \text{ for every } G \in \mathcal{G}(n, d)\ . \tag{3.6}$$

Since $\Delta_G = d$ for a d-regular graph, (3.5) follows from Proposition 3.2 via (3.6). □

Proof of Theorem 2.1. (a) Since $d = 2$, $\ell = 0$ follows from (3.2) if n is even, and from (3.1) if n is odd and $\alpha \leq (n/2) - 1$.

(b) Follows from Lemma 3.3.

(c) Given $d = 2$, every graph in $\mathcal{G}(n, 2)$ is a sum of vertex-disjoint cycles. Suppose $\gamma = 2$, so $n/3 - 1 < \alpha \leq n/2 - 1$. Then an even cycle has minimum interference 0, a 3-cycle has minimum interference 1, and an odd cycle with five or more vertices

has minimum interference 0 or 1. It follows that $L = M$ if $n = 3M$ (M 3-cycles), $L = M - 1$ if $n = 3M + 1$ ($M - 1$ 3-cycles, one 4-cycle), and $L \in \{M - 1, M\}$ if $n = 3M + 2$. The last case uses $M - 1$ 3-cycles and one 5-cycle. When the 5-cycle's vertices are colored successively as 1, $\alpha + 2$, $2\alpha + 3$, $n - 2\alpha - 1$, and $n - \alpha$, it has no interference if $[n-(2\alpha+3)]+[n-2\alpha-1] > \alpha$, i.e., if $\alpha < (2n-4)/5$, so $L = M-1$ in this case. More generally, suppose one vertex of the 5-cycle is colored 1. Its neighbors must have colors in $[\alpha + 2, n - \alpha]$ to avoid interference. Then their uncolored neighbors, which are adjacent, must have colors in $[2\alpha + 3, \ldots, n, 1, \ldots, n - 2\alpha - 1]$ to avoid interference. This set has $\max D = [n + (n - 2\alpha - 1)] - (2\alpha + 3)$, which is $\leq \alpha$ if $(2n - 4)/5 \leq \alpha$. Hence, $L = (M - 1) + 1$ for $n = 3M + 2$ if $(2n - 4)/5 \leq \alpha$. □

4. Minimal interference level: Theorem 2.2. We prove Theorem 2.2 in this section. The ranges stated where $\ell(n, \alpha, d) = 0$ follow from Lemma 3.1, so the main content of parts (a) and (b) of Theorem 2.2 concerns the values $\ell(n, \alpha, d)$ for $d > n/2$. To obtain these we use a variant of Turán's theorem (Turán [19]; Bondy and Murty [3, p. 110]), which we state as a lemma. An application of the lemma at the end of the section yields the exact value of $L(n, \alpha, n-1)$ as well as $\ell(n, \alpha, n-1)$. Recall that an *equi-t-partition* of a vertex set V is a partition $\{V_1, \ldots, V_t\}$ with $||V_i| - |V_j|| \leq 1$ for all $i, j \in \{1, \ldots, t\}$.

LEMMA 4.1. *The maximum number of noninterfering edges in the complete graph K_n with vertex set V and threshold parameter α is attained only by a coloring $f : V \to [n]$ that has $D(f(x),\ f(y)) > \alpha$ whenever x and y are in different parts of an equi-γ-partition of V.*

Proof. Suppose that a coloring f of the complete graph K_n has f_i vertices of color i and $f_i f_j > 0$ for some $i \neq j$ with $D(i, j) \leq \alpha$. Let m_{ab} denote the number of vertices of colors other than a and b that interfere with a and not b. If all color-i vertices are recolored j, the net increase in interference is $f_i(m_{ji} - m_{ij})$; if all color-j vertices are recolored i, the net increase in interference if $f_j(m_{ij} - m_{ji})$. Hence, at least one of the recolorings does not increase interference. Continuing this recoloring process implies that noninterference in K_n is maximized by a γ-partite partition of V such that $D(f(x),\ f(y)) > \alpha$ whenever x and y are in different parts of the partition. Turán's theorem then implies that maximum noninterference obtains only when the partition is an equi-γ-partition. □

We can assume without loss of generality that the coloring f found in Lemma 4.1 is constant on each part of an equi-γ-partition, with $f(V) = \{(i - 1)(\alpha + 1) + 1 : i = 1, \ldots, \gamma\}$. If interfering edges are then dropped from K_n, we obtain a complete equi-γ-partite graph with zero interference and chromatic number γ. This graph is regular if and only if γ divides n and each part of the partition has n/γ vertices.

Proof of Theorem 2.2. Throughout this proof $\gamma = 2$, so that

$$n/3 - 1 < \alpha \leq n/2 - 1\ . \tag{4.1}$$

We consider first (a) and (b). The ranges given where $\ell(n, \alpha, d) = 0$ come from Lemma 3.1. So assume now that $d > \frac{n}{2}$. Let G_0 be a complete bipartite graph $\{A, B\}$ such that

$$|A| = \lceil \frac{n}{2} \rceil \ \text{ and } \ |B| = \lfloor \frac{n}{2} \rfloor\ .$$

Lemma 4.1 implies that two-coloring G_0 using noninterfering colors for A and B uniquely maximizes the number of edges with no interference when $\gamma = 2$. Therefore $\ell \geq nd/2 - |A||B|$.

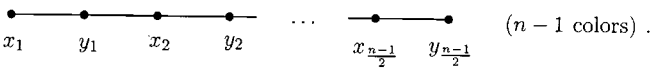
(a) Suppose n is even. If $n/2$ and d are odd, the number of edges needed within each part of G_0 to increase all degrees to d is $(n/2)(d-n/2)/2$, which is an integer since $d-n/2$ is even. It follows that if $n/2$ is even, or if $n/2$ and d are odd, then $\ell = (n/2)(d-n/2) = nd/2 - n^2/4$.

If instead $n/2$ is odd, and d is even, then $(n/2)(d-n/2)$ is odd, G_0 is not part of any graph in $\mathcal{G}(n,d)$, and $\ell > nd/2 - n^2/4$. We obtain $\ell = nd/2 - n^2/4 + 1$ by replacing G_0 with a complete bipartite graph G_1 with bipartition $\{A', B'\}$, $|A'| = n/2+1$ and $|B'| = n/2-1$. Beginning with G_1, each vertex in A' requires $d-n/2+1$ more degrees to have degree d, and each vertex in B' requires $d-n/2-1$ more edges added to have degree d. Both $d-n/2+1$ and $d-n/2-1$ are even, so edge additions as needed can be made within A' and B' to obtain $G \in \mathcal{G}(n,d)$. Therefore $\ell = nd/2 - (n/2+1)(n/2-1) = nd/2 - n^2/4 + 1$ in this case; and (2.9) is proved.

(b) Suppose n is odd, so d is even by (2.3). Beginning with G_0, each of the $(n+1)/2$ vertices in A requires $d-(n-1)/2$ more incident edges added to have degree d, and each of the $(n-1)/2$ vertices in B requires $d-(n+1)/2$ more incident edges added to have degree d. Each of $\{(n+1)/2, d-(n-1)/2\}$ and $\{(n-1)/2, d-(n+1)/2\}$ contains an even integer, so we can make the required additions of edges within A and B. Hence $\ell = nd/2 - [(n+1)/2][(n-1)/2] = nd/2 - (n^2-1)/4$, and (2.10) is proved.

It remains to prove (c), which has three parts, (i)–(iii). Assume henceforth that n is odd and $n-2\alpha < d < n/2$, with d even because n is odd. Augmented equi-bipartite graphs, illustrated at the top of Figure 2.1, show that $\ell \le d/2$ since they require $d/2$ edges within the $(n+1)/2$-vertex part to obtain degree d for every vertex. Sometimes $\ell = d/2$. A case in point is $\alpha = (n-3)/2$, the largest possible α for $\gamma = 2$ and odd n.

Suppose $\alpha = (n-3)/2$. Then $d > n - 2\alpha = 3 \Rightarrow d \in \{4, 6, \ldots, n-1\}$. Each vertex in the color set $[n]$ has exactly two others for which $D > \alpha$, and the graph of noninterfering colors is an n-cycle whose successive colors are $1, (n+3)/2, 2, (n+5)/2, 3, \ldots, (n+1)/2$. If every color were assigned to some vertex in $G \in \mathcal{G}(n,d)$, there would be at least $n(d-2)/2$ interference edges. But $n(d-2)/2 > d/2$, so f must avoid at least one color to attain ℓ. Deletion of one color from the n-cycle of noninterfering colors breaks the cycle and leaves the noninterference graph



Because all x_i colors interfere with each other, and all y_i colors interfere with each other, we can presume that f uses only one x_i and an adjacent y_j. This yields the augmented bipartite structure of the preceding paragraph, and it follows from maximization of between-parts edges that $\ell = d/2$. This completes the proof of (iii).

For (i) and (ii), assume $\alpha < (n-3)/2$ and consider an odd $r \ge 5$ sequence of colors $c_1, c_2, \ldots, c_r$ with $c_1 = 1$ and $D(c_{i+1}, c_i) \ge (\alpha+1)$ for $i = 1, \ldots, r-1$. The tightest such sequence has $c_i = (i-1)(\alpha+1)+1$ for $i = 2, \ldots, r$, where color $jn+k$, $1 \le k \le n$, is identical to color k. It follows that the final color c_r can be chosen not to interfere with $c_1 = 1 = jn+1$ if

$$\frac{1}{2}(r-1)n - (r-1)(\alpha+1) \ge (\alpha+1) \ ,$$

i.e., if

$$\alpha \le \left(\frac{r-1}{2r}\right) n - 1 \quad \Longleftrightarrow \quad r \ge \frac{n}{n - 2(\alpha+1)} \ . \tag{4.2}$$

We usually consider the smallest such odd $r \ge 5$ because this allows the $\ell = 0$ conclusion for the largest d values. Our approach, illustrated on the lower part of Figure 2.1, is to assign clumps of vertices to the c_i in such a way that all edges for $G \in \mathcal{G}(n,d)$ are between adjacent clumps on the noninterference color cycle $c_1, \ldots, c_r, c_1$.

Suppose (4.2) holds for a fixed odd $r \ge 5$. We assume that $r < n$ because the ensuing analysis requires this for $d \ge 3$. Let a and b be nonnegative integers that satisfy

$$n = ar + b, \quad 0 \le b < r \ .$$

We prove (i), then conclude with (ii). The analysis for (i) splits into three cases depending on the parity of a and $\lfloor r/4 \rfloor$.

Case 1: a odd.

Case 2: a even, $\lfloor r/4 \rfloor$ odd.

Case 3: a even, $\lfloor r/4 \rfloor$ even.

Because n is odd, Case 1 requires b to be even and Cases 2 and 3 require b to be odd.

Case 1. Given an odd a, we partition the n vertices into b clumps of $a+1$ vertices each and $r-b$ clumps of a vertices each. The clumps are assigned to colors in the noninterference cycle $c_1, \ldots, c_r, c_1$ so that the clumps of each type are contiguous. Cases for $b = 0$ and $b = 4$ are illustrated at the top of Figure 4.1. We begin at the central (top) a clump and proceed symmetrically in both directions around the color cycle, assigning between-clumps edges as we go so that all vertices end up with degree $2a$. The required edges into the next clump encountered are distributed as equally as possible to the vertices in that clump. When we get into the clumps with $a+1$ vertices, the number of between-clumps edges needed will generally be less than the maximum possible number of $(a+1)^2$. Numbers of between-clumps edges used to get degree $2a$ for every vertex are shown on the noninterference lines between the c_i on Figure 4.1.

The preceding construction yields $\ell(n,\alpha,d) = 0$ for $d = 2a = 2(n-b)/r$. If even d is less than $2a$, say $d = 2a'$ with $a' < a$, we modify the procedure by using fewer between-clumps edges for the required vertex degrees: clump sizes are unchanged. Because $n/r < d/2 = a$ yields the contradiction that $n < ra$, it follows for Case 1 that $\ell = 0$ if $d \le (2/r)n$.

Case 2. With a even and $\lfloor r/4 \rfloor$ odd, we have b odd and $r \in \{5, 7, 13, 15, 21, 23, \ldots\}$. In this case we assign $a-1$ vertices to c_1 and proceed in each direction around the c_i cycle, assigning a, $a+1$, a, $a-1$, a, $a+1$, a, $a-1, \ldots, a^0$, a^* vertices to the next $(r-1)/2$ c_i in order. The penultimate number a^0 equals a if $r \in \{5, 13, 21, \ldots\}$ and is $a+1$ if $r \in \{7, 15, 23, \ldots\}$. The ultimate number a^* in chosen so that there are $[n-(a-1)]/2$ vertices (excluding the $a-1$ for c_1) on each side of the color cycle. If $a^0 = a$ then $a^* = a + (b+1)/2$, and if $a^0 = a+1$ then $a^* = a + (b-1)/2$. The two cases are shown on the lower left of Figure 4.1 with numbers of between-clumps edges that give degree $d = 2a$ for every vertex. If even d is less than $2a$, fewer edges are used, as needed, down the two sides. As in Case 1, we get $\ell = 0$ if $d \le (2/r)n$.

Case 3. With a even and $\lfloor r/4 \rfloor$ even, we have b odd and $r \in \{9, 11, 17, 19, 25, 27, \ldots\}$. Here we assign $a+1$ vertices to c_1 and proceed with a, $a-1$, a, $a+1$, a, $a-1$, a, $a+1, \ldots, a^0$, a^* vertices assigned to the next $(r-1)/2$ c_i in each direction away

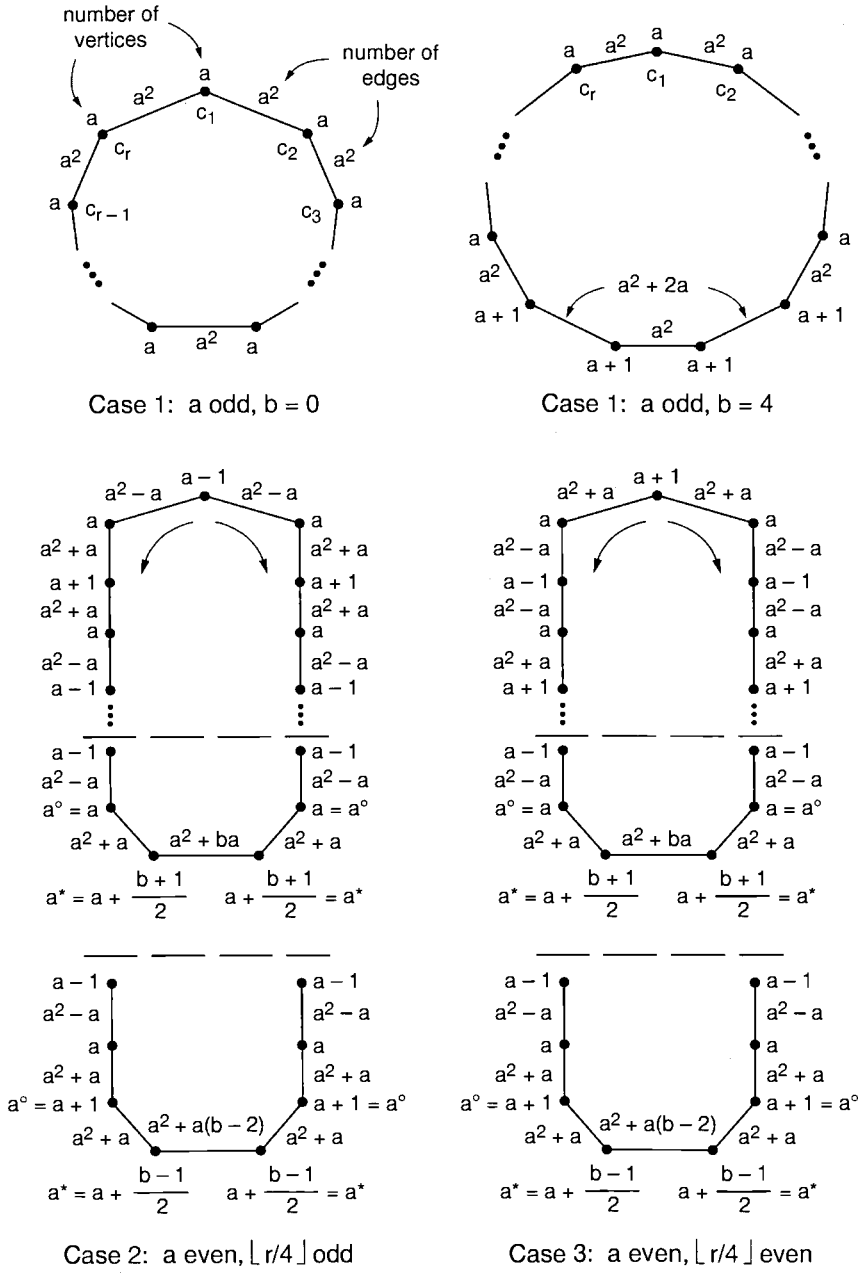


FIG. 4.1.

from c_1. We get $a^0 = a$ and $a^* = a + (b+1)/2$ if $r \in \{9, 17, 25, \ldots\}$, and $a^0 = a + 1$ and $a^* = a + (b-1)/2$ if $r \in \{11, 19, 27, \ldots\}$. The two cases are shown on the lower right of Figure 4.1. As before, $\ell = 0$ if $d \leq (2/r)n$.

This completes the proof of (i), after defining s by $r = 2s + 1$. We have also checked that the construction used here cannot yield $l = 0$ unless the conditions of (i) hold.

There is, however, one other set of circumstances where this construction yields a value of $\ell < d/2$ for some $d > (2/r)n$, and these circumstances are exactly the hypotheses of (ii), namely

$$\text{(4.3)} \qquad \begin{cases} d & \geq & 8, \ \ d/2 \ \text{ is even}, \\ r & = & 4s+1 \ \text{ for some integer } s \geq 1, \\ n & = & r(d/2)-1 \ \ (n \text{ is odd since } d/2 \text{ is even}), \\ \alpha & & \text{satisfies (4.2)} \ . \end{cases}$$

In this case we partition the vertices into $(r+1)/2$ clumps of $d/2-1$ vertices each and $(r-1)/2$ clumps of $d/2+1$ vertices each. The clumps (-1 for $d/2-1$, $+1$ for $d/2+1$) are arranged around the noninterference color cycle $c_1, c_2, \ldots, c_r$ as $-1,-1,-1,+1,+1,-1,-1,+1,+1,\ldots,-1,-1,+1,+1$. We use all possible between-clumps edges. This gives degree d for every vertex except those in the c_2 clump, which has $2(d/2-1)^2 = d^2/2 - 2d + 2$ incoming edges from c_1 and c_3. The degree total for c_2 should be $d(d/2-1) = d^2/2 - d$, so we need to add $(d-2)/2 = d/2-1$ edges within c_2 to get degree d for each c_2 vertex. Prior to the additions, each c_2 vertex has degree $d-2$ by our equalization construction, so the additions can be made by a complete cycle within the clump. It follows that $\ell \leq d/2-1$, proving (ii). □

We conclude this section by noting that the modified Turán's theorem (Lemma 4.1) easily allows us to completely settle the case of degree $d = n-1$.

COROLLARY 4.2. For $d = n-1$,

$$\text{(4.4)} \qquad \ell(n,\alpha,n-1) = L(n,\alpha,n-1) = \lfloor \frac{n}{\gamma} \rfloor \left(n - \frac{1}{2}\gamma \left(\lfloor \frac{n}{\gamma} \rfloor + 1 \right) \right) \ .$$

Proof. Write

$$n = p\gamma + q \ , \quad 0 \leq q < \gamma \ ,$$

so $p = \lfloor \frac{n}{\gamma} \rfloor$. An equi-$\gamma$-partition of an n vertex set has

$$\begin{cases} q \text{ parts, each with } p \ + \ 1 \text{ vertices}, \\ \\ \gamma - q \text{ parts, each with } p \text{ vertices}. \end{cases}$$

Now the unique graph $G \in \mathcal{G}(n, n-1)$ is K_n, so applying Lemma 4.1, we have

$$\ell(n,\alpha,n-1) = L(n,\alpha,n-1) = q \begin{pmatrix} p+1 \\ 2 \end{pmatrix} + (\gamma - q) \begin{pmatrix} p \\ 2 \end{pmatrix} \ ,$$

which is (4.4). □

5. Chromatic number bound: Theorem 1.1. This section gives a self-contained proof of Theorem 1.1. We first recall two preliminary facts, stated as propositions.

PROPOSITION 5.1 (see Dirac [7]). *Let G be a simple graph. If every vertex of G is of degree at least $|V(G)|/2$, then G is Hamiltonian; that is, G has a cycle of length $|V(G)|$.*

Proof. For the proof, see Bondy and Murty [3, p. 54]. □

Recall that a *matching* in a simple graph G is a subset of mutually vertex-disjoint edges of G. A matching is *perfect* if every vertex in G is on some edge of the matching. The following is a consequence of a well-known theorem of Hall [13].

PROPOSITION 5.2 (Marriage Theorem). *If G is a d-regular bipartite graph with $d > 0$, then G has a perfect matching.*

Proof. For the proof, see Bondy and Murty [3, p. 73]. □

We study the function $\phi(n, d; r)$ defined by

$$\phi(n,d;r) = \begin{cases} 1 & \text{if there exists an } n\text{-vertex } d\text{-regular } r\text{-colorable graph,} \\ \\ 0 & \text{otherwise .} \end{cases}$$

When $\phi(n, d; r) = 1$ we let $G(n, d; r)$ denote such a d-regular r-colorable (that is, r-partite) graph having n vertices. We consider only values in which nd is even.

Our first observation is that because an r-colorable graph is also $(r+1)$-colorable,

$$\phi(n,d;r_1) \le \phi(n,d;r_2) \quad \text{if } r_1 < r_2 \ . \tag{5.1}$$

The purpose of the next two lemmas is to prove that $\phi(n, d; r)$ is monotone when $r \ge 3$ is held fixed and d varies over values where nd is even.

LEMMA 5.3. (a) *If $d \le n/2$ and if either $r \ge 3$ or $r = 2$ and n is even, then*

$$\phi(n,d;r) = 1 \ . \tag{5.2}$$

(b) *If $d \ge n/2$, then*

$$\phi(n,d;r) = 1 \ \ \textit{implies} \ \ \phi(n,d-2;r) = 1 \ . \tag{5.3}$$

If in addition n is even, then

$$\phi(n,d;r) = 1 \ \ \textit{implies} \ \ \phi(n,d-1;r) = 1 \ . \tag{5.4}$$

Proof. (a) Suppose that n is even. The inequality (5.1) implies that it is enough to show

$$\phi(n,d;2) = 1 \text{ for } d \le \frac{n}{2}, \qquad n \text{ even} \ . \tag{5.5}$$

We use reverse induction on $d \le n/2$. For the base case $d = n/2$, the complete equi-2-partite graph gives $\phi(n, n/2; 2) = 1$. For the induction step, suppose we know that $\phi(n, d; 2) = 1$. Then a d-regular bipartite graph $G(n, d; 2)$ exists, and by Proposition 5.2 it has a perfect matching M. Remove all edges in M from G to obtain a $(d-1)$-regular bipartite graph $G(n, d-1; 2)$. Hence $\phi(n, d-1, 2) = 1$.

Suppose n is odd. Then (5.1) implies that it is enough to show

$$\phi(n,d;3) = 1 \text{ for } d \le \frac{n}{2}, \qquad n \text{ odd} \ . \tag{5.6}$$

Now d must be even by (2.3), and $d \le (n-1)/2$. Because $n-1$ is even, we have $\phi(n-1, d; 2) = 1$ by (5.5). Consider $G := G(n-1, d; 2)$ with $|V(G)| = n-1$. By Proposition 5.2 we may find a perfect matching of G, say $M = \{\{x_1, y_1\}, \ldots, \{x_k, y_k\}\}$, with $k = (n-1)/2 \ge d/2$. Remove from G the edges $\{x_1, y_1\}, \ldots, \{x_{\frac{d}{2}}, y_{\frac{d}{2}}\}$, and add to G a new vertex z and the edges $\{z, x_i\}$ and $\{z, y_i\}$ for $1 \le i \le d/2$. Then it is easy to see that the resulting graph is a d-regular 3-partite graph with n vertices, which proves (5.6).

(b) Let $G = G(n, d; r)$, which exists by hypothesis. Since $d \geq n/2$, Proposition 5.1 guarantees that G has a Hamiltonian cycle C. Removing all edges from C yields a $G(n, d-2; r)$, so $\phi(n, d-2; r) = 1$. If moreover n is even, then C has even length and we get a perfect matching M by taking alternate edges in C. Removing all edges in M from G yields a $G(n, d-1; r)$, so $\phi(n, d-1; r) = 1$ in this case. □

LEMMA 5.4. *If $r \geq 3$, then*

$$\phi(n, d_1; r) \geq \phi(n, d_2; r) \quad if \quad d_1 < d_2 \ , \tag{5.7}$$

provided that nd_1 and nd_2 are both even.

Proof. Suppose $d_1 \leq n/2$. Then by Lemma 5.3(a), $\phi(n, d_1; r) = 1$ for all $r \geq 3$, and we are done.

Suppose $d_1 > n/2$. For even n, Lemma 5.3(b) used inductively on decreasing d gives

$$\phi(n, d_2; r) = 1 \Rightarrow \phi(n, d_2 - 1; r) = 1 \Rightarrow \ldots \Rightarrow \phi(n, d_1; r) = 1 \ .$$

For odd n, since nd_1 and nd_2 are both even, both d_1 and d_2 must be even. Now Lemma 5.3(b) gives

$$\phi(n, d_2; r) = 1 \Rightarrow \phi(n, d_2 - 2; r) = 1 \Rightarrow \ldots \Rightarrow \phi(n, d_1; r) = 1 \ ,$$

so (5.7) follows. □

Proof of Theorem 1.1. To commence the proof, we define p and q by

$$n = pr + q \quad \text{with} \quad 0 \leq q < r \ , \tag{5.8}$$

that is, $p = \lfloor n/r \rfloor \geq 1$. Note that r divides $n+1$ if and only if $q = r - 1$. In terms of p and q the assertions of the theorem then become

(i) If $q = 0$, then $\phi(n, d; r) = 1$ if $d \leq n - p$.

(ii) If $1 \leq q \leq r - 2$, then $\phi(n, d; r) = 1$ if $d \leq n - p - 1$.

(iii) If $q = r - 1$ then $\phi(n, d; r) = 1$ if $d \leq n - p - 2$.

To prove (i)–(iii), we use the complete equi-r-partite graph $G^r(n)$ defined as follows. The graph $G^r(n)$ has vertices $V = \{v_1, v_2, \ldots, v_n\}$, and for $1 \leq j \leq r$ we define the vertex sets

$$X_j = \{v_i : i \equiv j \pmod{r}\} \ . \tag{5.9}$$

The edge set of $G^r(n)$ is

$$E(G^r(n)) = \{\{v_i, v_j\} : i \not\equiv j (\mathrm{mod}\ r)\} \ .$$

Here $\{X_1, \ldots, X_r\}$ is an equi-r-partition of V with

$$|X_1| = |X_2| = \cdots = |X_q| = p + 1, \ |X_{q+1}| = \cdots = |X_r| = p \ . \tag{5.10}$$

For $1 \leq a \leq b \leq r$ we let $G^r_{a,b}$ denote the induced subgraph of $G^r(n)$ on the vertex set

$$V_{a,b} := \cup_{j=a}^{b} X_j \ .$$

To prove (i), if $q = 0$ then $G^r(n)$ is an $(n-p)$-regular graph, hence

$$\phi(n, n - p; r) = 1 \ . \tag{5.11}$$

Lemma 5.4 implies $\phi(n,d;r)=1$ if $d\le n-p$, and (i) follows.

To prove (ii), let $H=G^r_{q+1,r}$. Then (5.10) shows that H is a $p(r-q-1)$-regular graph having $p(r-q)$ vertices. Now $r-q\ge 2$ implies that H has degree $p(r-q-1)$, which is greater than half its vertices, so H has a Hamiltonian cycle C by Proposition 5.1.

If $p(r-q)$ is even, then H has a perfect matching M obtained by taking every other edge in C. Removing all edges in M from $G^r(n)$, the resulting graph is $(n-p-1)$-regular, hence $\phi(n,n-p-1;r)=1$. Lemma 5.4 then completes the proof of (ii).

If $p(r-q)$ is odd, then p is odd, hence so is

$$n=pr+q=(p+1)q+p(r-q)\ .$$

Then $n-p-1$ is also odd, so $d=n-p-1$ is forbidden by (2.3). Thus it suffices to show that $\phi(n,n-p-2;r)=1$ in this case, for then Lemma 5.4 gives $\phi(n,d;r)=1$ for $d\le n-p-2$.

Let $H':=G^r_{1,q}$. Then H' is a $(p+1)(q-1)$-regular graph with $(p+1)q$ vertices. If $q>1$, then

$$(p+1)(q-1)\ge (p+1)q/2\ ,$$

hence H' is Hamiltonian. Since $(p+1)q$ is even, H' has a perfect matching M'. Removing all edges in $M'\cup C$ from $G^r(n)$, the resulting graph is $(n-p-2)$-regular, hence $\phi(n,n-p-2;r)=1$.

Suppose $q=1$. Notice that since $p(r-1)$ is odd, $r\ne 3$, hence $r\ge 4$. Let H'' be the induced subgraph of $G^r(n)$ on the set

$$\{v_{jr+2}\in X_2:(p+1)/2\le j\le p\}\cup\bigcup_{j=3}^{r}X_j,$$

and

$$E:=\{\{v_{ir+1},v_{jr+2}\}:0\le j\le (p-1)/2,\ i=2j \text{ or } i=2j+1\}\ .$$

Then the number of vertices of H'' is $p(r-2)+(p-1)/2$ and the minimum degree of H'' is $p(r-3)+(p-1)/2$. Since

$$p(r-3)+(p-1)/2\ge\frac{1}{2}(p(r-2)+(p-1)/2) \text{ for } r\ge 4\ ,$$

Proposition 5.1 implies that H'' has a Hamiltonian cycle C''. By removing all edges in $C''\cup E$ from $G^r(n)$ we have an $(n-p-2)$-regular graph, hence $\phi(n,n-p-2;r)=1$.

To prove (iii) we proceed by induction on r, with an induction step from r to $r+2$. There are two base cases, $r=3$ and $r=4$.

Base Case $r=3$. We have $q=2$, so $n=3p+2$. Let

$$E_1=\{\{v_{3i},v_{3i-2}\}:i=1,2,\dots,p\} \text{ and } E_2=\{\{v_{3i},v_{3i-1}\}:i=1,2,\dots p\}\ .$$

Consider the graph G obtained by removing from $G^3(n)$ all edges in $E_1\cup E_2\cup\{v_{3p+1},v_{3p+2}\}$. Then it is easy to see that G is $(n-p-2)$-regular, so $\phi(n,n-p-2;r)=1$. Now Lemma 5.4 gives $\phi(n,d;r)=1$ for $d\le n-p-2$.

Base Case $r=4$. We have $q=3$, and $n=4p+3$. Suppose first that p is odd. We relabel the vertices of $G^4(n)$ so that the sets X_j in (5.9) become

$$X_j=\{w_i:i\equiv j \pmod 3\} \text{ for } j=1,2,3, \text{ while } X_4=\{u_i:1\le i\le p\}\ . \tag{5.12}$$

Let H be the subgraph of $G^4(n)$ induced on the vertex set $\{w_j : 2p+1 \le j \le 3p+3\}$. Then $|V(H)| = p+3$ is even and H is Hamiltonian. Thus H has a perfect matching, call it M. Consider the edge set

$$E = \{\{u_i, w_j\} : 1 \le i \le p,\ j = 2i-1 \text{ or } 2i\} ,$$

and form a graph G by removing all edges in $E \cup M$ from $G^4(n)$. Then G is an $(n-p-2)$-regular subgraph of $G^4(n)$, hence $\phi(n, n-p-2; r) = 1$, and $\phi(n, d; r) = 1$ for $d \le n-p-2$ by Lemma 5.4.

Suppose now that p is even. Then $n = 4p+3$ is odd and $n-p-2$ is also odd, so $d = n-p-2$ is forbidden by (2.3). It suffices, therefore, to show that $\phi(n, n-p-3; r) = 1$ in this case, for then Lemma 5.4 gives $\phi(n, d; r) = 1$ for $d \le n-p-3$, hence also for $d \le n-p-2$. We use the vertex labelling (5.12), and let H be the subgraph of $G^4(n)$ induced on $\{w_j : 1 \le j \le 3p\}$. Then $|V(H)| = 3p$ is even, and H is Hamiltonian, so H has a perfect matching M. Consider the edge set

$$E = \{\{u_i, w_j\} : 1 \le i \le p,\ j = 3i-2,\ 3i-1 \text{ or } 3i\}$$
$$\cup \{\{w_{3p+1}, w_{3p+2}\},\ \{w_{3p+2}, w_{3p+3}\},\ \{w_{3p+3},\ w_{3p+1}\}\} .$$

Form a graph G by removing $E \cup M$ from $G^4(n)$. It is an $(n-p-3)$-regular graph, whence $\phi(n, n-p-3; r) = 1$.

Induction Step. Fix $r \ge 5$ and define

$$d_0 := d_0(n, r) = \max\{d : d \le n-p-2,\ nd \text{ is even}\} ,$$

so $d_0 = n-p-2$ or $n-p-3$. It is enough to show that $\phi(n, d_0; r) = 1$, for Lemma 5.4 then yields $\phi(n, d; r) = 1$ for $d \le n-p-2, nd$ even.

To do this, set

$$n' = n - 2(p+1) = p(r-2) + q - 2 ,$$

where $q = r-1$ so $q-2 > 0$. Then $d_1 = d_0(n, r) - n'$ has $0 \le d_1 \le p$, and furthermore we easily check that

$$d' := d_0(n', r-2) = d_0(n, r) - 2(p+1) . \tag{5.13}$$

Take $r' = r-2$, whence $q' = q-2 = r'-1$. We may apply the induction hypothesis at $r' = r-2$ to conclude that there exists a d'-regular $(r-2)$-partite graph $G = G(n', d'; r-2)$. Let H be a d_1-regular bipartite graph with $2(p+1)$ vertices disjoint from those of G; such a graph H exists by Lemma 5.3(a). Take the disjoint union of G and H and add in all edges between $V(G)$ and $V(H)$ to obtain a new graph G' on n vertices which is $d_0(n, r)$-regular, according to (5.13). Thus $\phi(n, d_0; r) = 1$, completing the induction step for (iii). □

6. Minimal interference level: Theorems 2.3 and 2.4. In this section we study the range $\gamma \ge 3$ and prove Theorems 2.3 and 2.4. The cases where $\ell(n, \alpha, d) = 0$, i.e., for d smaller than about $n-p$, follow from Theorem 1.1 applied with $r = \gamma$. For the remaining cases, the harder step in the proofs is obtaining the (exact) lower bounds for $\ell(n, \alpha, d)$. The upper bounds are obtained by explicit construction.

We proceed to derive a lower bound for $\ell(n, \alpha, d)$ stated as Lemma 6.2 below. Let G be any d-regular graph on n-vertices, let $f : V(G) \to \{1, 2, \ldots, n\}$ be a given coloring of G, and let α also be given. We begin by partitioning the n colors into

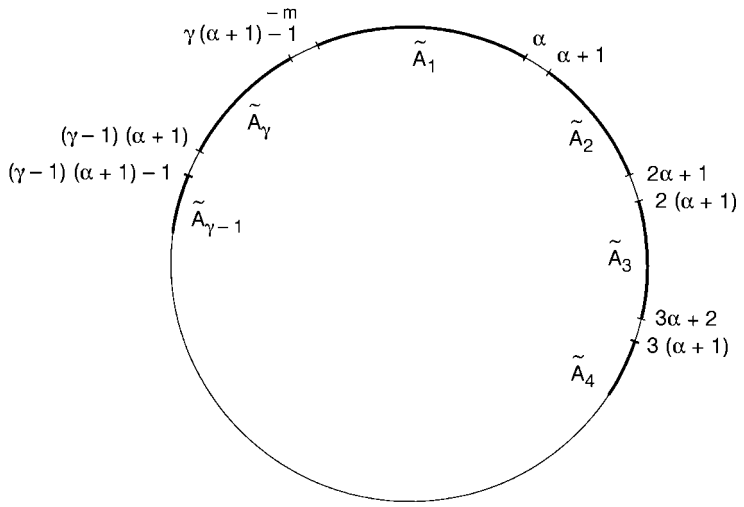


FIG. 6.1. *Color set partition.*

γ groups $\{\tilde{A}_i : 1 \le i \le \gamma\}$, such that each group $\tilde{A}_i$ consists of consecutive colors and the groups $\tilde{A}_1, \ldots, \tilde{A}_\gamma$ are themselves consecutively arranged with respect to the cyclic ordering of colors (mod n), with all groups but $\tilde{A}_1$ containing exactly $\alpha + 1$ colors, and $\tilde{A}_1$ contains the remaining $\alpha + 1 + m$ colors. Here m is given by

$$n = \gamma(\alpha + 1) + m, \text{ with } 0 \le m < \alpha + 1 \ , \tag{6.1}$$

and such a partition is completely determined by the choice of $\tilde{A}_1 = \{i, i+1, \ldots, i+\alpha+1+m\}$. We now choose $\tilde{A}_1$ so as to minimize the number of vertices v in G that are assigned colors $f(v)$ in $\tilde{A}_1$. After doing this, we have the freedom to cyclically relabel the colors (via the map $\phi_\ell(j) = j + \ell (\text{mod } n)$) without affecting which edges have vertex colors that interfere. We use this freedom to specify that

$$\tilde{A}_1 := \{-m, -m+1, \ldots, \alpha - 1, \alpha\} \ ,$$

in which case

$$\tilde{A}_i := \{j : (i-1)(\alpha+1) \le j < i(\alpha+1)\} \quad \text{for} \quad 2 \le i \le \gamma \ :$$

(see Figure 6.1). Notice that for $2 \le i \le \gamma$ any two colors in $\tilde{A}_i$ interfere with each other.

This partition of the colors induces a corresponding partition of the vertices of G into the color classes

$$A_i := \{v \in G : f(v) \in \tilde{A}_i\}, \qquad 1 \le i \le \gamma \ . \tag{6.2}$$

Now set

$$a_i := |A_i| \ .$$

We now count the edges in G and in its complement $\bar{G} = K_n - G$ in various ways. For any two subsets V and W of vertices, let $e(V,W)$ count the number of edges between vertices in V and those in W, and let $\bar{V} := V(G) \setminus V$. Let $\bar{a}_{i,j}$ count the number of edges between A_i and A_j that are *not* in G, which is

$$\bar{a}_{i,j} := a_i a_j - e(A_i, A_j) \ , \qquad 1 \le i,\ j \le \gamma \ .$$

Along with this we define

$$\bar{a}_i := \Sigma_{j \ne i} \bar{a}_{i,j} = a_i(n - a_i) - e(A_i, \tilde{A}_i) \ , \qquad 1 \le i \le \gamma \ .$$

The d-regularity of G then yields

$$e(A_i, A_i) = \frac{1}{2}(da_i - e(A_i, \tilde{A}_i)) = \frac{1}{2}(a_i(d + a_i - n) + \bar{a}_i) \ . \tag{6.3}$$

The *potential interfering edge set* $B_{i,j}$ between vertices in A_i and those in A_j is

$$B_{i,j} := B_{i,j}(G,f) = \{\{v,w\} \in E(K_n) : v \in A_i, w \in A_j, \text{with } D(f(v), f(w)) \le \alpha\} \ .$$

The *actual interfering edge set* is $E(G) \cap B_{i,j}$ and we set

$$c_{i,j} := |E(G) \cap B_{i,j}| \ .$$

We clearly have

$$\bar{a}_{i,j} + c_{i,j} \ge |B_{i,j}| \ . \tag{6.4}$$

Finally, let δ^* and δ count the potential and actual noninterfering edges in A_1, respectively, i.e.,

$$\begin{aligned} \delta^* &:= |\{\{v,w\} \in E(K_n) : v,w \in A_1 \text{ and } D(f(v), f(w)) \ge \alpha + 1\}| \ , \\ \delta &:= |\{\{v,w\} \in E(G) : v,w \in A_1 \text{ and } D(f(v), f(w)) \ge \alpha + 1\}| \ . \end{aligned}$$

Certainly $\delta^* \ge \delta$. Since all edges between the vertices in the same component A_i interfere, except for δ edges in A_1, we obtain the bound

$$\begin{aligned} I_\alpha(G,f) &= \Sigma_{i<j} c_{i,j} + \Sigma_{i=1}^{\gamma} e(A_i, A_i) - \delta \\ &\ge \Sigma_{i<j} c_{i,j} + \Sigma_{i=1}^{\gamma} e(A_i, A_i) - \delta^* \ . \end{aligned} \tag{6.5}$$

To bound this further, we need the following bounds for edges connecting a vertex in the color set $\tilde{A}_1$ to a vertex in its two neighboring color sets $\tilde{A}_2$ and $\tilde{A}_\gamma$.

LEMMA 6.1. *We have*

$$\bar{a}_{1,2} + c_{1,2} \ge \delta^* \ , \tag{6.6}$$

and

$$\bar{a}_{1,\gamma} + c_{1,\gamma} \ge \delta^* \ . \tag{6.7}$$

Proof. We start with (6.6). By (6.4) it is enough to show that

$$|B_{1,2}| \ge \delta^* \ .$$

It suffices to show for fixed $v \in A_1$ with $\alpha + 1 - m \le f(v) \le \alpha$ that

$$|\{w \in A_2 : D(f(v), f(w)) \le m\}| \ge |\{w \in A_1 : D(f(v), f(w)) \ge \alpha + 1\}| \ , \tag{6.8}$$

because, using $\alpha \ge m$, this implies that, for sums over $v \in A_1$ with $\alpha + 1 - m \le f(v) \le \alpha$,

$$\begin{aligned} |B_{1,2}| &\ge \Sigma_v |\{w \in A_2 : D(f(v), f(w)) \le m\}| \\ &\ge \Sigma_v |\{w \in A_1 : D(f(v), f(w)) \ge \alpha + 1\}| = \delta^* \ . \end{aligned}$$

To prove (6.8), given $v \in A_1$ with $\alpha + 1 - m \le f(v) \le \alpha$, we define the vertex set

$$A' := \{w \in V(G) : f(w) \in \{f(v) - \alpha, f(v) - \alpha + 1, \ldots, f(v) + m\}\} \subseteq A_1 \cup A_2 \ .$$

This is a set of $\alpha + 1 + m$ consecutive colors, hence $|A'| \ge |A_1|$ by the minimizing property of the color set $\tilde{A}_1$. Now $\alpha + 1 - m \le f(v) \le \alpha$ implies that

$$A_1 \cap A' = \{w \in V(G) : f(w) \in \{f(v) - \alpha, \ldots, \alpha\}\} \ .$$

Thus

$$|A' \setminus (A_1 \cap A')| \ge |A_1 \setminus (A_1 \cap A')|,$$

which is exactly (6.8). Thus (6.6) follows.

The proof of (6.7) is analogous. □

To state the lower bound lemma, recall that the quantities p and q are defined by

$$n = p\gamma + q \quad \text{with } 0 \le q < \gamma \ ,$$

so $p = \lfloor n/\gamma \rfloor$.

LEMMA 6.2. *If $d \ge n - p$ then*

$$\ell(n, \alpha, d) \ge q\lceil (p+1)(d+p+1-n)/2 \rceil + (\gamma - q)\lceil p(d+p-n)/2 \rceil \ . \tag{6.9}$$

Proof. We derive this result from the general bound

$$I_\alpha(G, f) \ge \Sigma_{i=1}^{\gamma} \lceil a_i(d + a_i - n)/2 \rceil \ , \tag{6.10}$$

where $a_i = |A_i|$ for the vertex partition (6.2). To establish (6.10), we first note that Lemma 6.1 yields

$$\frac{1}{2}(\bar{a}_{1,2} + \bar{a}_{1,\gamma}) + c_{1,2} + c_{1,\gamma} \ge \delta^* \ .$$

Together with (6.3), this yields

$$\begin{aligned} e(A_1, A_1) - \delta^* + c_{1,2} + c_{1,\gamma} &\ge e(A_1, A_1) - \tfrac{1}{2}(\bar{a}_{1,2} + \bar{a}_{1,\gamma}) \\ &\ge \tfrac{1}{2} a_1(d - a_1 - n) \ . \end{aligned}$$

Since the left side of this inequality is an integer,

$$e(A_1, A_1) - \delta^* + c_{1,2} + c_{1,\gamma} \ge \lceil a_1(d - a_1 - n)/2 \rceil \ .$$

However, (6.3) also gives

$$e(A_i, A_i) \geq \lceil a_i(d + a_i - n)/2 \rceil, \quad \text{for } 2 \leq i \leq \gamma \ .$$

Substituting these bounds in (6.5) yields (6.10).

To derive (6.9), we minimize the right side of (6.10) over all possible values: $a_i \geq 0$ subject to $\Sigma_{i=1}^{\gamma} a_i = n$. It is easy to verify that this occurs when all the a_i's are as equal as possible, i.e.,

$$\text{(6.11)} \qquad \begin{cases} q \text{ of the } a_i \text{ take the value } p+1, \\ \\ \gamma - q \text{ of the } a_i \text{ take the value } p \ . \end{cases}$$

Thus

$$I_\alpha(G, f) \geq q\lceil (p+1)(d+p+1-n)/2 \rceil + (\gamma - q)\lceil p(d+p-n)/2 \rceil \ ,$$

which gives (6.9). □

Proof of Theorem 2.3. (a) This bound follows from Theorem 1.1, taking $r = \gamma$ noting that $q = 0$ guarantees that r doesn't divide $n + 1$.

(b) For $d > n - p$ we first establish the lower bounds

$$\text{(6.12)} \qquad \ell(n, \alpha, d) \geq \frac{n(d - n + p)}{2} + \mu$$

where

$$\text{(6.13)} \qquad \mu = \begin{cases} 0 & \text{if } n-d \text{ is odd or if } n-d \text{ is even} \\ & \text{and } p \text{ is even}, \\ \\ \frac{\gamma}{2} & \text{if } n-d \text{ is even and } p \text{ is odd} \ , \end{cases}$$

using Lemma 6.2. The case $q = 0$ is $n = p\gamma$, so (6.9) simplifies to

$$\begin{aligned} \ell(n, \alpha, d) &\geq \gamma\lceil p(d - n + p)/2 \rceil \\ &= \tfrac{n}{p}\lceil p(d - n + p)/2 \rceil. \end{aligned}$$

Now (6.12) follows on determining the cases for which $p(d - n + p)$ is odd.

To show that this bound is attained, we simply construct the graph G with the coloring f that makes (6.11) hold. The constructions are easy and are left to the reader. □

Proof of Theorem 2.4. (a) The bounds where $\ell(n, \alpha, d) = 0$ follow from Theorem 1.1 with $r = \gamma$.

There remains the case in which $d = n - p - 1$ and $q = \gamma - 1$, i.e., when $n = \lfloor n/\gamma \rfloor \gamma + \gamma - 1$ (where Theorem 1.1 does not apply). We must show that

$$\ell(n, \alpha, n - p - 1) = \frac{p}{2} \ .$$

For the upper bound $\ell \leq p/2$, it suffices to construct an appropriate graph. Note first that p must be even since if p is odd, then $n = p\gamma + \gamma - 1 \equiv (p+1)\gamma - 1$ is odd and $d = n - p - 1$ is also odd, contradicting (2.3). Now consider the equi-γ-partite graph $G^\gamma(n)$ defined in the proof of Theorem 1.1. We take a perfect

matching M from the induced subgraph of $G^\gamma(n)$ on the vertex set $(X_{\gamma-1} \setminus \{v_n\}) \cup X_\gamma$. We remove all the edges in M from $G^\gamma(n)$ and add the edges $\{v_{\gamma-1},\ v_{2\gamma-1}\}$, $\{v_{3\gamma-1}, v_{4\gamma-1}\}, \ldots, \{v_{(p-1)\gamma-1},\ v_{p\gamma-1}\}$. Then it is straightforward to check that the resulting graph G is $(n-p-1)$-regular and it clearly has exactly $p/2$ interfering edges when the sets X_i are colored with γ mutually noninterfering colors.

To show the lower bound $\ell \geq p/2$, let G be an $(n-p-1)$-regular graph and f an n-coloring of $V(G)$ such that

$$I_\alpha(G, f) = \ell(n, \alpha, n-p-1) \ .$$

Take the partition $\{A_i : 1 \leq i \leq \gamma\}$ of $V(G)$ associated to f constructed at the beginning of this section. We consider cases.

Case (i). $a_1 \geq p+2$.
The minimality property of A_1 implies that, for all $v \in V(G)$,

$$|\{w \in V(G) : f(w) = f(v) + j \pmod n \text{ with } -m \leq j \leq \alpha\}| \geq p+2 \ .$$

Since $d = n-p-1$, for each $v \in V(G)$ there exists $w \in V(G) \setminus \{v\}$ such that $|f(v) - f(w)| \leq \alpha$. Thus $I_\alpha(G, f) \geq n/2 > p/2$.

Case (ii). $a_i \geq p+2$ for some $2 \leq i \leq \gamma$.
Here the equality in (6.5) combined with $e(A_1, A_1) \geq \delta$ yields

$$I_\alpha(G, f) \geq \Sigma_{i=2}^\gamma e(A_i, A_i) \ . \tag{6.14}$$

Using (6.3) we then have

$$I_\alpha(G, f) \geq e(A_i, A_i) \geq \frac{1}{2} a_i(d + a_i - n) \geq \frac{p+2}{2} > \frac{p}{2} \ .$$

Case (iii). All $a_i \leq p+1$.
Since $n = (p+1)\gamma - 1$, this case requires that $\gamma - 1$ of the a_i equal $p+1$ and one a_i equals p.

Suppose first that $a_1 = p$. Observe that (6.14) and (6.3) yield

$$\begin{aligned} I_\alpha(G, f) &\geq \Sigma_{i=2}^\gamma e(A_i, A_i) \\ &\geq \tfrac{1}{2}\Sigma_{i=2}^\gamma \bar{a}_i \geq \tfrac{1}{2}\Sigma_{i=2}^\gamma \bar{a}_{i,1} \ . \end{aligned} \tag{6.15}$$

Now (6.2) and $a_1 = p$ give

$$\begin{aligned} \Sigma_{i=2}^\gamma \bar{a}_{i,1} &= \bar{a}_1 = (a_1(n-a_1) - e(A_1, \bar{A}_1)) \\ &\geq p(n-p) - pd = p \ . \end{aligned}$$

Substituting this in (6.15) gives $I_\alpha(G, f) \geq p/2$.

Suppose finally that $a_1 = p+1$. Since $\gamma \geq 3$, and only one $a_i = p$, either $a_2 = p+1$ or $a_\gamma = p+1$ or both. We treat only the case that $a_2 = p+1$, since the argument for $a_\gamma = p+1$ is similar. Let $a_{i_0} = p$. Now by (6.5) and (6.3)

$$\begin{aligned} I_\alpha(G, f) &\geq \tfrac{1}{2}(\bar{a}_1 + \bar{a}_2) + c_{1,2} - \delta^* + \tfrac{1}{2}\sum_{\substack{i=3 \\ i \neq i_0}}^\gamma \bar{a}_i \\ &\geq \tfrac{1}{2}(2\bar{a}_{1,2} + \bar{a}_{1,i_0} + \bar{a}_{2,i_0}) + c_{1,2} - \delta^* + \tfrac{1}{2}\sum_{\substack{i=3 \\ i \neq i_0}}^\gamma \bar{a}_i \ . \end{aligned}$$

Lemma 6.1 gives $\bar{a}_{1,2} + c_{1,2} \geq \delta^*$, hence

$$\begin{aligned} I_\alpha(G,f) &\geq \tfrac{1}{2}(\Sigma_{i \neq i_0} \bar{a}_{i,i_0}) = \tfrac{1}{2}\bar{a}_{i_0} \\ &\geq \tfrac{1}{2}(p(n-p) - pd) = \tfrac{p}{2} \ , \end{aligned}$$

completing case (iii).

(b) We start from the formula (6.9) of Lemma 6.2, which gives a lower bound. We claim that equality occurs. This formula of $\ell(n,\alpha,d)$ splits into several cases, according to when $(p+1)(d+p+1-n)/2$ and $p(d+p-n)/2$ are integers or half-integers, and consideration of the parities of $n-d$ and p leads to the formulas for θ in (2.12).

For the upper bound, obtaining equality in the formula for $\ell(n,\alpha,d)$ requires (6.11) to hold, and this easily determines the construction of a suitable graph G and a coloring f. We omit the details. □

7. Minimax interference level: Theorems 2.5 and 2.6. We conclude by proving the bounds for $L(n,\alpha,d)$ stated in section 2.

Proof of Theorem 2.5. To show part (a), the condition $L(n,\alpha,d) = 0$ certainly holds if the chromatic number $\chi_G \leq \gamma$ for all $G \in \mathcal{G}(n,d)$. This holds for $\gamma > d$ by Brooks' theorem (Proposition 3.2). For the case $\gamma = d \geq 3$ we use the strong version of Brooks' theorem, which states that $\chi_G \leq \Delta_G$ if no component of G is an odd cycle or a complete subgraph. Here $\Delta_G = d$, and $d \geq 3$ implies there are no odd cycles, while the condition $n < 2(d+1)$ prohibits any connected component being the complete subgraph K_d, for any other components must be d-regular but have at most d vertices, a contradiction.

To show part (b), suppose that $n \geq 2(d+1)$. Let $G \in \mathcal{G}(n,d)$ consist of a complete graph K_{d+1} plus a d-regular graph G' on the other $n-(d+1) \geq d+1$ vertices. If d is odd then n is even, so that $n-(d+1)$ is even, and the existence of G' is assured by a theorem of Erdős and Gallai [8] for simple graphs with specified degree sequences. If $\gamma \leq d$, at least two vertices of K_{d+1} interfere, so $L > 0$.

Suppose that $n = 2(d+1-a)$, $a \geq 1$. This implies $d \geq 2a$ because we presume that $n \geq d+2$. Let G consist of two disjoint copies of K_{d+1-a}, adding edges between them that increase every degree to d. Each vertex requires a such edges, and this is feasible because $d+1-a > a$. If $\gamma \leq d-a$, at least two vertices of K_{d+1-a} interfere, so $L > 0$.

Suppose finally that $n = 2(d+1-a)+1$, $a \geq 1$. Then n is odd, so d must be even. Moreover, $n \geq d+2 \Rightarrow n \geq d+3 \Rightarrow d \geq 2a$. Let G consist of two disjoint graphs $G_1 = K_{d+1-a}$ and $G_2 = K_{d+2-a}$ with edge additions and deletions as follows. Add a edges from each G_1 vertex to G_2 vertices in as equal a way as possible for resulting vertex degrees in G_2. Then each vertex in G_1 has degree d, x vertices in G_2 have degree $d+1$, and y vertices in G_2 have degree d, where

$$\begin{aligned} x + y &= d + 2 - a \ , \\ xa + y(a-1) &= a(d+1-a) \ . \end{aligned}$$

These equations imply that $x = d+2-2a > 0$, so x is even. We then remove $x/2$ edges within G_2 so that all vertices have degree d. We thus arrive at a graph $G \in \mathcal{G}(n,d)$. If $\gamma \leq d-a$ then at least two vertices in G_1 interfere, so $L > 0$. Thus part (b) holds. □

Proof of Theorem 2.6. Suppose that $3 \le d \le n-1$ and that $\gamma \le d$. Let P, Q, U, and W be nonnegative integers that satisfy

$$\begin{aligned} d+1 &= P\gamma + Q, \quad 0 \le Q < \gamma \ ; \\ n &= (d+1)U + W, \quad 0 \le W < d+1 \ . \end{aligned}$$

To derive the upper bound on L in Theorem 2.6, let G be any graph in $\mathcal{G}(n,d)$. Let S denote the family of all partitions of the vertex set of G into γ groups, with q groups of size $p+1$ and $\gamma - q$ groups of size p. We adopt a probability model for S that assigns probability $1/|S|$ to each partition. Whichever partition obtains, we use γ mutually noninterfering colors for the γ groups in the partition. Suppose $\{u,v\}$ is an edge in G. The probability that u and v lie in the same part of a member of S, so that $\{u,v\}$ is an interference edge, is

$$\frac{q\binom{p+1}{2} + (\gamma - q)\binom{p}{2}}{\binom{n}{2}} = \frac{n(n-\gamma) + q(\gamma - q)}{\gamma n(n-1)} \ .$$

The expected number $E[I]$ of interference edges is $nd/2$ times this amount, i.e.,

$$E[I] = \frac{d(n(n-\gamma) + q(\gamma - q))}{2\gamma(n-1)} \ ,$$

so some member of S has a coloring that gives less than or equal to $E[I]$ edges whose vertices interfere. This is true for every $G \in \mathcal{G}(n,d)$. Therefore we get the upper bound

$$L(n,\alpha,d) \le \frac{d}{2\gamma(n-1)}[n(n-\gamma) + q(\gamma - q)] \ .$$

For the lower bound, assume initially that $(d+1)$ divides n, so $W = 0$ and $U = n/(d+1)$. Let G consist of U disjoint copies of K_{d+1}. Then $L(n,\alpha,d) \ge UL(d+1,\alpha)$, where $L(d+1,\alpha)$ is the minimum number of interfering edges in K_{d+1} for an $f : V_{d+1} \to [n]$. The analysis in Lemma 4.1 shows that $L(d+1,\alpha)$ is attained by an equi-γ-partition of V_{d+1} with f constant in each part. Since an equi-γ-partition of V_{d+1} has

$$\begin{cases} Q \text{ groups of } P+1 \text{ vertices each,} \\ \gamma - Q \text{ groups of } P \text{ vertices each,} \end{cases}$$

we have

$$\begin{aligned} L(d+1,\alpha) &= [Q(P+1)P + (\gamma - Q)P(P-1)]/2 \\ &= \frac{(d+1)(d+1-\gamma) + Q(\gamma - Q)}{2\gamma} \ . \end{aligned}$$

Since $L(n,\alpha,d) \ge UL(d+1,\alpha)$, this gives

$$L(n,\alpha,d) \ge \frac{n(d+1-\gamma)}{2\gamma} + \frac{nQ(\gamma - Q)}{2\gamma(d+1)} \ ,$$

when $d+1$ divides n.

Suppose $(d+1)$ does not divide n. Let $n = (d+1)U + W$, where $U = \lfloor n/(d+1) \rfloor$ and $0 < W \leq d$. To form G we begin with U disjoint copies of K_{d+1} and a disjoint K_W. Each vertex in K_W needs $d-(W-1)$ more incident edges, so we add a total of $W(d+1-W)$ edges between K_W and the K_{d+1} in such a way that $W(d+1-w)/2$ edges can be removed from within the K_{d+1} to end up with degree d for every vertex. Note that $W(d+1-W)$ is even, for otherwise both n and d would be odd. We ignore possible interference within K_W and allow for the possibility that every edge removed from the K_{d+1} is an interference edge to get the lower bound

$$\begin{aligned} L(n,\alpha,d) &\geq UL(d+1,\alpha) - W(d+1-W)/2 \\ &= \frac{\lfloor n/(d+1) \rfloor}{2\gamma}[(d+1)(d+1-\gamma) + Q(\gamma - Q)] - \frac{W(d+1-W)}{2} \ . \end{aligned}$$ □

REFERENCES

[1] M. Benveniste, M. Bernstein, A. Greenberg, N. J. A. Sloane, J. Tung, and P. E. Wright, *Lattices, adjacent channel interference and dynamic channel allocation in cellular systems*, 1995, in preparation.

[2] M. Bernstein, N. J. A. Sloane, and P. E. Wright, *On sublattices of the hexagonal lattice*, Discrete Math., 170 (1997), pp. 29–39.

[3] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland, New York, 1976.

[4] I. Bonias, *T-colorings of Complete Graphs*, Ph.D. thesis, Northeastern University, Boston, MA, 1991.

[5] R. L. Brooks, *On coloring the nodes of a network*, Proc. Cambridge Phil. Soc., 37 (1941), pp. 194–197.

[6] M. B. Cozzens and F. S. Roberts, *T-colorings of graphs and the channel assignment problem*, Congr. Numer., 35 (1982), pp. 191–208.

[7] G. A. Dirac, *Some theorems on abstract graphs*, Proc. London Math. Soc., 2 (1952), pp. 69–81.

[8] P. Erdős and T. Gallai, *Graphs with prescribed degrees of vertices*, in Hungarian, Mat. Lapok, 11 (1960), pp. 264–274.

[9] J. R. Griggs and D. D.-F. Liu, *The channel assignment problem for mutually adjacent sites*, J. Combin. Theory A, 68 (1994), pp. 169–183.

[10] D. R. Guichard, *No-hole k-tuple $(r+1)$-distant colorings*, Discrete Appl. Math., 64 (1996), pp. 87–92.

[11] D. R. Guichard and J. W. Krussel, *Pair labellings of graphs*, SIAM J. Discrete Math., 5 (1992), pp. 144–149.

[12] W. K. Hale, *Frequency assignment: Theory and application*, Proc. IEEE, 68 (1980), pp. 1497–1514.

[13] P. Hall, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26–30.

[14] D.-F. Liu, *Graph Homomorphisms and the Channel Assignment Problem*, Ph.D. thesis, University of South Carolina, Columbia, SC, 1991.

[15] A. Raychaudhuri, *Further results on T-coloring and frequency assignment problems*, SIAM J. Discrete Math., 7 (1994), pp. 605–613.

[16] F. S. Roberts, *T-colorings of graphs: recent results and open problems*, Discrete Math., 93 (1991), pp. 229–245.

[17] B. A. Tesman, *List T-colorings of graphs*, Discrete Appl. Math., 45 (1993), pp. 277–289.

[18] D. S. Troxell, *No-hole k-tuple $(r+1)$-distant colorings of odd cycles*, Discrete Appl. Math., 64 (1996), pp. 67–85.

[19] P. Turán, *An extremal problem in graph theory*, in Hungarian, Mat. Fiz. Lapok, 48 (1941), pp. 436–452.

[20] A. Vince, *Star chromatic number*, J. Graph Theory, 12 (1988), pp. 551–559.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 41–53, February 1998

003

COMBINATORIAL PROPERTIES AND CONSTRUCTIONS OF TRACEABILITY SCHEMES AND FRAMEPROOF CODES*

D. R. STINSON† AND R. WEI‡

Abstract. In this paper, we investigate combinatorial properties and constructions of two recent topics of cryptographic interest, namely frameproof codes for digital fingerprinting and traceability schemes for broadcast encryption. We first give combinatorial descriptions of these two objects in terms of set systems and also discuss the Hamming distance of frameproof codes when viewed as error-correcting codes. From these descriptions, it is seen that existence of a c-traceability scheme implies the existence of a c-frameproof code. We then give several constructions of frameproof codes and traceability schemes by using combinatorial structures such as t-designs, packing designs, error-correcting codes, and perfect hash families. We also investigate embeddings of frameproof codes and traceability schemes, which allow a given scheme to be expanded at a later date to accommodate more users. Finally, we look briefly at bounds which establish necessary conditions for existence of these structures.

Key words. traceability scheme, frameproof code, t-design, hash family

AMS subject classifications. 94A60, 05B05, 05B15, 05B40

PII. S0895480196304246

1. Introduction.

Traceability schemes for broadcast encryption were defined by Chor, Fiat, and Naor [8], and frameproof codes for digital fingerprinting were proposed by Boneh and Shaw [4]. Although these two objects were designed for different purposes, they have some similar aspects. One of the purposes of this paper is to investigate the relations between traceability schemes and frameproof codes. We first give combinatorial descriptions of these two objects in terms of set systems and also discuss the Hamming distance of frameproof codes when viewed as error-correcting codes. From these descriptions, it is seen that existence of a c-traceability scheme implies the existence of a c-frameproof code.

In [4, 8], some constructions of frameproof codes and traceability schemes were provided. We will provide new (explicit) constructions by using combinatorial structures such as t-designs, packing designs, error-correcting codes, and perfect hash families. We also investigate embeddings of frameproof codes and traceability schemes, which allow a given scheme to be expanded at a later date to accommodate more users. Finally, we look briefly at bounds which establish necessary conditions for existence of these structures.

In this rest of this section we review the definitions of c-frameproof codes and c-traceability schemes which were given in [4] and [8], respectively.

1.1. Frameproof codes.

In order to protect a product (such as computer software, for example), a distributor marks each copy with some codeword and then ships each user his data marked with that codeword (for some examples of how this might be done in practice, see [5]). This marking allows the distributor to detect any unauthorized copy and trace it back to the user. Since a marked object can be traced, the

*Received by the editors May 20, 1996; accepted for publication (in revised form) January 30, 1997. This research was supported by NSF grant CCR-9402141.
http://www.siam.org/journals/sidma/11-1/30424.html
†Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588 (stinson@bibd.unl.edu).
‡Department of Mathematics and Statistics, University of Nebraska-Lincoln, Lincoln, NE 68588 (rwei@cse.unl.edu).

users will be deterred from releasing an unauthorized copy. However, a coalition of users may detect some of the marks, namely the ones where their copies differ. They can then change these marks arbitrarily. To prevent a group of users from "framing" another user, Boneh and Shaw [4] defined the concept of c-frameproof codes. A c-frameproof code has the property that no coalition of at most c users can frame a user not in the coalition.

Let v and b be positive integers (b denotes the number of users in the scheme). A set $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(b)}\} \subseteq \{0,1\}^v$ is called a (v,b)-*code*, and each $w^{(i)}$ is called a *codeword*. So a codeword is a binary v-tuple. We can use a $b \times v$ matrix M to depict a (v,b)-code, in which each row of M is a codeword in Γ.

Let Γ be a (v,b)-code. Suppose $C = \{w^{(u_1)}, w^{(u_2)}, \ldots, w^{(u_d)}\} \subseteq \Gamma$. For $i \in \{1,2,\ldots,v\}$, we say that bit position i is *undetectable* for C if

$$w_i^{(u_1)} = w_i^{(u_2)} = \cdots = w_i^{(u_d)}.$$

Let $\mathcal{U}(C)$ be the set of undetectable bit positions for C. Then

$$F(C) = \{w \in \{0,1\}^v : w|_{\mathcal{U}(C)} = w^{(u_i)}|_{\mathcal{U}(C)} \text{ for all } w^{(u_i)} \in C\}$$

is called the *feasible set* of C. (If $\mathcal{U}(C) = \emptyset$, then we define $F(C) = \{0,1\}^v$.) The feasible set $F(C)$ represents the set of all possible v-tuples that could be produced by the coalition C by comparing the d codewords they jointly hold. If there is a codeword $w^{(j)} \in F(C) \backslash C$, then user j could be "framed" if the coalition C produces the v-tuple $w^{(j)}$. The following definition from [4] is motivated by the desire for this situation not to occur.

DEFINITION 1.1. *A (v,b)-code Γ is called a* c-frameproof code *if, for every $W \subseteq \Gamma$ such that $|W| \leq c$, we have $F(W) \cap \Gamma = W$. We will say that Γ is a c-FPC(v,b) for short.*

Thus, in a c-frameproof code the only codewords in the feasible set of a coalition of at most c users are the codewords of the members of the coalition. Hence, no coalition of at most c users can frame a user who is not in the coalition.

Example 1.1 (see [4]). For any integer b, there exists a b-FPC(b,b). The matrix depicting the code is a $b \times b$ identity matrix. □

Example 1.2. There exists a 2-FPC$(3,4)$. The matrix depicting the code is as follows:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \qquad \square$$

1.2. Traceability schemes. In many situations, such as a pay-per-view television broadcast, the data is only available to authorized users. To prevent an unauthorized user from accessing the data, the data supplier will encrypt the data and give the authorized users keys to decrypt it. Some unauthorized users (*pirate users*) might obtain some decryption keys from a group of one or more authorized users (called *traitors*). Then the pirate users can decrypt data that they are not entitled to. To prevent this, Chor, Fiat and Naor [8] devised a traitor tracing scheme, called a *traceability scheme*, which will reveal at least one traitor on the confiscation of a pirate decoder.

Suppose there are a total of b users. The data supplier generates a base set T of v keys and assigns k keys to each user. These k keys comprise a user's *personal key*, and we will denote the personal key for user U by $P(U)$. A *message* consists of an enabling block and a cipher block. A *cipher block* is the encryption of the actual plaintext data using some secret key S. The *enabling block* consists of data, which is encrypted using some or all of the v keys in the base set, the decryption of which will allow the recovery of S. Every authorized user should be able to recover S using his or her personal key and then decrypt the cipher block using S to obtain the plaintext data.

Some traitors may conspire and give an unauthorized user a "pirate decoder," F. The pirate decoder F will consist of k base keys, chosen from T, such that $F \subseteq \cup_{U \in \mathcal{C}} P(U)$, where $\mathcal{C}$ is the coalition of traitors. An unauthorized user may be able to decrypt S using a pirate decoder F. The goal of the data supplier is to assign keys to the users in such a way that when a pirate decoder is captured and the keys it possesses are examined, it should be possible to detect at least one traitor in the coalition $\mathcal{C}$, provided that $|\mathcal{C}| \leq c$ (where c is a predetermined threshold).

Traitor detection would be done by computing $|F \cap P(U)|$ for all users U. If $|F \cap P(U)| \geq |F \cap P(V)|$ for all users $V \neq U$, then U is defined to be an *exposed user*.

DEFINITION 1.2. *Suppose any exposed user U is a member of the coalition $\mathcal{C}$ whenever a pirate decoder F is produced by $\mathcal{C}$ and $|\mathcal{C}| \leq c$. Then the scheme is called a* c-traceability scheme *and it is denoted by* c-$TS(k, b, v)$.

Let us now briefly discuss the difference between our scheme and that of [8]. In [8], $v = nk$ for some integer n, and the set T of base keys is partitioned into k subsets S_i, each of size n. We will denote $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,n}\}$, $1 \leq i \leq k$. Each personal key $P(U)$ is a *transversal* of $(S_1, \ldots, S_k)$ (i.e., it contains exactly one key from each S_i). Suppose the secret key S is chosen from an abelian group G. To encrypt S, the data supplier splits S into k shares $r_1, r_2, \ldots, r_k \in G$ such that $\sum r_i = S$. Then, for $1 \leq i \leq n$, he encrypts every share r_i with each of the n keys in S_i by computing $t_{i,j} = r_i + s_{i,j}$. The nk values $t_{i,j}$ comprise the enabling block. Each authorized user has one key from S_i, so he or she can decrypt every r_i, and thus compute S.

In our definition, we do not require that each personal key be a transversal. A personal key can be made up of any selection of k base keys from the set T. The data supplier can use a k out of v threshold scheme (such as the Shamir scheme [13], for example) to construct v shares of the key S and then encrypt each share r_i with the key s_i, for every $s_i \in T$.

Note that our definition is a generalization of the one given in [8]. However, the generalization has to do with the way that the enabling block is formed, and not with the traceability property of the scheme. Our definition of the traceability property is the same as in [8].

Example 1.3. We present a 2-TS$(5, 21, 21)$. The set of base keys is $\mathbb{Z}_{21}$. The personal key for user i $(0 \leq i \leq 20)$ is

$$P(i) = \{3 + i, 6 + i, 7 + i, 12 + i, 14 + i\},$$

where all arithmetic is done in $\mathbb{Z}_{21}$. (This is an application of a construction we will present in Theorem 3.6.) It can be shown that any two base keys occur together in exactly one personal key. Now, consider what happens when two traitors U and V construct a pirate decoder, F. The pirate decoder F must contain at least three personal keys from $P(U)$ or $P(V)$. However, for any other user $W \neq U, V$, $|F \cap P(W)| \leq 2$. Hence either U or V will be the exposed user if the pirate decoder F is examined. □

1.3. Previous results. In the construction of frameproof codes and traceability schemes, the main goal is to accommodate as many users as possible. In other words, we want to find constructions with b as large as possible, given values for the parameters c and v (and k, in the case of traceability schemes). In general, we would prefer explicit constructions for these objects as opposed to nonconstructive existence results.

For example, Boneh and Shaw [4] proved the following interesting result.

THEOREM 1.3. *For any integers $c, v > 0$, there exists a c-FPC$(v, 2^{v/(16c^2)})$.*

However, as noted in [4], the proof is not constructive. Hence, they also provide an explicit construction for a c-FPC$(v, 2^{\sqrt{v}/c})$.

Similarly, Chor, Fiat, and Naor [8] gave an interesting nonconstructive existence result for traceability schemes, as follows.

THEOREM 1.4. *For any integers $c, v > 0$, there exists a c-TS$(v/(2c^2), 2^{v/(8c^4)}, v)$.*

We will provide several explicit constructions for frameproof codes and traceability schemes later in this paper. Although our constructions may not be as good asymptotically as those in [4] and [8], they will often be better for relatively small values of c and v. (For example, in order to obtain $b \geq 2$ in Theorem 1.3, it is necessary to take $v \geq 16c^2$, so the construction is not useful for small values of v.) As well, our constructions are very simple and could be implemented very easily and efficiently.

2. Combinatorial descriptions. In this section, we give combinatorial descriptions of c-frameproof codes and c-traceability schemes. From these descriptions, it is fairly easy to see that the existence of a c-TS(k, b, v) implies the existence of a c-FPC(v, b).

We will use the terminology of set systems. A *set system* is a pair $(X, \mathcal{B})$ where X is a set of elements called *points* and $\mathcal{B}$ is a set of subsets of X, the members of which are called *blocks*. A set system can be described by an incidence matrix. Let $(X, \mathcal{B})$ be a set system where $X = \{x_1, x_2, \ldots, x_v\}$ and $\mathcal{B} = \{B_1, B_2, \ldots, B_b\}$. The *incidence matrix* of $(X, \mathcal{B})$ is the $b \times v$ matrix $A = (a_{ij})$, where

$$a_{ij} = \begin{cases} 1 & \text{if } x_j \in B_i, \\ 0 & \text{if } x_j \notin B_i. \end{cases}$$

Conversely, given an incidence matrix, we can define an associated set system in an obvious way.

2.1. Description of c-frameproof codes. Since a c-FPC(v, b) is a $b \times v$ $(0,1)$-matrix, we can view a frameproof code as an incidence matrix or as a set system, as defined above. We have the following characterization of frameproof codes as set systems.

THEOREM 2.1. *There exists a c-FPC(v, b) if and only if there exists a set system $(X, \mathcal{B})$ such that $|X| = v, |\mathcal{B}| = b$ and for any subset of $d \leq c$ blocks $B_1, B_2, \ldots, B_d \in \mathcal{B}$, there does not exist a block $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$ such that*

$$\bigcap_{i=1}^{d} B_i \subseteq B \subseteq \bigcup_{i=1}^{d} B_i.$$

Proof. Suppose $w^{(1)}, w^{(2)}, \ldots, w^{(d)}$ are d codewords in a c-FPC(v, b) $(d \leq c)$. Without loss of generality, assume that in these codewords the first s bit positions are 0, the next t bit positions are 1, and in every other bit position at least one of the d codewords has the value 0 and at least one has the value 1. (Hence, the undetectable

bit positions are the first $s+t$ bit positions.) Then it is not hard to see that the frameproof property is equivalent to saying that any other codeword w has at least one 1 in the first s bit positions or at least one 0 in the next t bit positions. In other words, there does not exist a codeword with 0's in the first s bit positions and 1's in the next t bit positions.

Suppose $B_1, B_2, \ldots, B_d$ are the blocks in the set system corresponding to the d codewords $w^{(1)}, w^{(2)}, \ldots, w^{(d)}$. Then

$$\bigcap_{i=1}^{d} B_i = \{x_{s+1}, \ldots, x_{s+t}\},$$

and

$$\bigcup_{i=1}^{d} B_i = \{x_{s+1}, \ldots, x_v\}.$$

Hence the frameproof condition is equivalent to saying that there does not exist a block B such that $\cap B_i \subseteq B \subseteq \cup B_i$. □

2.2. Description of c-traceability schemes. Since a c-TS(k, b, v) consists of b k-subsets of a v-set, we can think of it as a set system, where X is the set of base keys and $\mathcal{B}$ is the set of personal keys.

THEOREM 2.2. *There exists a c-TS(k, b, v) if and only if there exists a set system $(X, \mathcal{B})$ such that $|X| = v, |\mathcal{B}| = b$, and $|B| = k$ for every $B \in \mathcal{B}$, with the property that for every choice of $d \leq c$ blocks $B_1, B_2, \ldots, B_d \in \mathcal{B}$ and for any k-subset $F \subseteq \cup_{j=1}^{d} B_j$, there does not exist a block $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$ such that $|F \cap B_j| \leq |F \cap B|$ for $1 \leq j \leq d$.*

Proof. Suppose $(X, \mathcal{B})$ is a c-TS(k, b, v). For every set of $d \leq c$ personal keys $B_1, B_2, \ldots, B_d \in \mathcal{B}$, for any k-subset $F \subseteq \cup_{j=1}^{d} B_j$ (i.e., a pirate decoder) and for any other personal key B, there exists a B_j $(1 \leq j \leq d)$ such that $|F \cap B_j| > |F \cap B|$. So there is no block $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$ satisfying $|F \cap B_j| \leq |F \cap B|$ for $1 \leq j \leq d$. The converse is also straightforward. □

2.3. Relationship of traceability schemes and frameproof codes. We prove the following theorem relating traceability schemes and frameproof codes.

THEOREM 2.3. *If there exists a c-TS(k, b, v), then there exists a c-FPC(v, b).*

Proof. Let $(X, \mathcal{B})$ be the set system corresponding to a c-TS(k, b, v). We prove that $(X, \mathcal{B})$ is a c-FPC(v, b). Suppose not; then there exist $d \leq c$ blocks, $B_1, B_2, \ldots, B_d \in \mathcal{B}$, and a block $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$ such that $B \subseteq \cup_{i=1}^{d} B_i$. Then $|B \cap B_j| \leq |B \cap B|$ for $1 \leq j \leq d$. But this contradicts Theorem 2.2 (letting $F = B$). □

2.4. Hamming distance of 2-frameproof codes. Now we investigate some properties of the Hamming distance of c-frameproof codes. For any (v, b)-code, let $d(x, y)$ denote the Hamming distance of two codewords x, y.

Denote

$$d_{max} = \max\{d(w^{(i)}, w^{(j)}) : w^{(i)}, w^{(j)} \in \Gamma, i \neq j\},$$

and

$$d_{min} = \min\{d(w^{(i)}, w^{(j)}) : w^{(i)}, w^{(j)} \in \Gamma, i \neq j\}.$$

THEOREM 2.4. *A (v, b)-code Γ is 2-frameproof if and only if*

$$d(w^{(i)}, w^{(j)}) < d(w^{(i)}, w^{(h)}) + d(w^{(h)}, w^{(j)}),$$

for all $i \neq j \neq h \neq i$.

Proof. Let $w^{(i)}, w^{(j)}$ and $w^{(h)}$ be any three distinct codewords. Without loss of generality, assume that $U(\{w^{(i)}, w^{(j)}\}) = \{1, \ldots, r\}$, so the first r bits of $w^{(i)}$ and $w^{(j)}$ are the same.

We have that $d(w^{(i)}, w^{(j)}) = v - r$. Since Hamming distance is a metric, we have that

$$d(w^{(i)}, w^{(h)}) + d(w^{(h)}, w^{(j)}) \geq v - r.$$

Now, it will be the case that

$$d(w^{(i)}, w^{(h)}) + d(w^{(h)}, w^{(j)}) > d(w^{(i)}, w^{(j)})$$

if and only if there is at least one bit position within the first r bit positions such that $w^{(h)}$ is different from $w^{(i)}$ and $w^{(j)}$. But this is just the condition that the code is 2-frameproof (as stated in the proof of Theorem 2.1). □

The following result is an immediate corollary of the previous lemma.

COROLLARY 2.5. *A (v, b)-code Γ is 2-frameproof if $d_{max} < 2d_{min}$.*

We give an example to illustrate the application of this corollary. In [6], a simple explicit construction is given for a $(q, (q^2 - q)/2)$-code with $d_{max} \leq q/2 + 3\sqrt{q}/2$ and $d_{min} \geq q/2 - 3\sqrt{q}/2$, for any prime power q. Hence, for $q > 81$, we see that $d_{max} < 2d_{min}$. In fact, we have verified by computer that $d_{max} < 2d_{min}$ for the codes produced by this construction for all odd prime powers q such that $31 \leq q \leq 79$. Applying Corollary 2.5, we obtain the following result.

THEOREM 2.6. *For any odd prime power $q \geq 31$, there exists a 2-FPC$(q, (q^2 - q)/2)$.*

3. Constructions from combinatorial structures. In this section, we will give some constructions of frameproof codes and traceability schemes from certain combinatorial designs, including t-designs, packing designs, and orthogonal arrays. All the results on design theory that we require can be found in standard references such as the *CRC Handbook of Combinatorial Designs* [9].

3.1. Constructions using t-designs. First we give the definition of a t-design.

DEFINITION 3.1. *A t-(v, k, λ) design is a set system $(X, \mathcal{B})$, where $|X| = v$, $|B| = k$ for every $B \in \mathcal{B}$, and every t-subset of X occurs in exactly λ blocks in $\mathcal{B}$.*

Note that, by simple counting, the number of the blocks in a t-$(v, k, 1)$ design is $b = \binom{v}{t}/\binom{k}{t}$. We will use t-$(v, k, 1)$ designs to construct frameproof codes and traceability schemes, as described in the following theorems.

THEOREM 3.2. *If there exists a t-$(v, k, 1)$ design, then there exists a c-FPC$(v, \binom{v}{t}/\binom{k}{t})$, where $c = \lfloor (k-1)/(t-1) \rfloor$.*

Proof. Denote $c = \lfloor (k-1)/(t-1) \rfloor$. Let $B_1, B_2, \ldots, B_d$ be $d \leq c$ distinct blocks, and let $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$. If $B \subseteq \cup_{i=1}^{d} B_i$, then there exists a B_i, where $1 \leq i \leq d$, such that $|B \cap B_i| \geq t$. Since we have a t-design with $\lambda = 1$, it follows that $B = B_i$, a contradiction. Hence, for any $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$, we have that $B \not\subseteq \cup_{i=1}^{d} B_i$. The t-design is a set system satisfying the conditions of Theorem 2.1, so the conclusion follows. □

Similarly, we can construct traceability schemes from t-$(v,k,1)$ designs; the value of c obtained is smaller, however.

THEOREM 3.3. *If there exists a t-$(v,k,1)$ design, then there exists a c-TS$(k, \binom{v}{t}/\binom{k}{t}, v)$, where $c = \left\lfloor \sqrt{(k-1)/(t-1)} \right\rfloor$.*

Proof. Suppose there exists a t-$(v,k,1)$ design $(X, \mathcal{B})$. Let $B_1, B_2, \ldots, B_d$ be $d \leq c$ distinct blocks. Let $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$. If $F \subseteq \cup_{i=1}^{d} B_i$, where $|F| = k$, then there exists a B_i, where $1 \leq i \leq d$, such that

$$\begin{aligned} |F \cap B_i| &\geq \left\lceil \frac{k}{c} \right\rceil \\ &\geq k\sqrt{\frac{t-1}{k-1}} \\ &> \sqrt{(k-1)(t-1)}. \end{aligned}$$

On the other hand, since $|B \cap B_j| \leq t-1$ for $1 \leq j \leq c$, we have

$$\begin{aligned} |B \cap F| &\leq c(t-1) \\ &\leq \sqrt{(k-1)(t-1)}. \end{aligned}$$

Hence, it follows that $|F \cap B_i| > |F \cap B|$. This shows that the t-design is a set system satisfying the conditions of Theorem 2.2, and the conclusion follows. □

There are many known results on existence and construction of t-$(v,k,1)$ designs for $t = 2, 3$. On the other hand, no t-$(v,k,1)$ design with $v > k > t$ is known to exist for $t \geq 6$. However, known infinite classes of 2- and 3-designs provide some nice infinite classes of frameproof codes and traceability schemes. We illustrate with a few samples of typical results that can be obtained.

First, for $3 \leq k \leq 5$, a 2-$(v,k,1)$ design exists if and only if $v \equiv 1$ or $k \bmod (k^2-k)$ (see [9, Chapter I.2]. Hence, we obtain the following theorem.

THEOREM 3.4. *There exist frameproof codes as follows.*

1. *There exists a 2-FPC$(v, v(v-1)/6)$ for all $v \equiv 1, 3 \bmod 6$.*
2. *There exists a 3-FPC$(v, v(v-1)/12)$ for all $v \equiv 1, 4 \bmod 12$.*
3. *There exists a 4-FPC$(v, v(v-1)/20)$ for all $v \equiv 1, 5 \bmod 20$.*

Similarly, we have the following theorem about the existence of 2-traceability schemes (note that to get $c \geq 2$ when $t = 2$ in Theorem 3.3, we need $k \geq 5$).

THEOREM 3.5. *There exists a 2-TS$(5, v(v-1)/20, v)$, for all $v \equiv 1, 5 \bmod 20$.*

A 2-$(q^2+q+1, q+1, 1)$ design is known as a *projective plane* of order q; such a design exists whenever q is a prime power (see [9, Chapter VI.7]). In a projective plane we have $b = v$, so the frameproof codes obtained from it are not interesting (in view of Example 1.1, which does better). However, the traceability schemes will be of interest.

THEOREM 3.6. *There exists a $\lfloor \sqrt{q} \rfloor$-TS$(q+1, q^2+q+1, q^2+q+1)$, for all prime powers q.*

Example 1.3 is in fact obtained from the case $q = 4$ of Theorem 3.6.

We give another class of examples derived from 3-$(q^2+1, q+1, 1)$ designs (these designs are called *inversive planes* and exist if q is a prime power; see [9, Chapter VI.7]).

THEOREM 3.7. *For any prime power q, there exists a $\lfloor \frac{q}{2} \rfloor$-FPC$(q^2+1, q^3+q)$ and a $\lfloor \sqrt{\frac{q}{2}} \rfloor$-TS$(q+1, q^3+q, q^2+1)$.*

3.2. Constructions using packing designs. Another type of combinatorial design which can be used to construct frameproof codes and traceability schemes are packing designs. We give the definition as follows.

DEFINITION 3.8. *A t-(v,k,λ)* packing design *is a set system $(X,\mathcal{B})$, where $|X| = v$, $|B| = k$ for every $B \in \mathcal{B}$, and every t-subset of X occurs in at most λ blocks in $\mathcal{B}$.*

Using the same argument as in the proof of Theorem 3.2, we have the following construction for frameproof codes.

THEOREM 3.9. *If there exists a t-$(v,k,1)$ packing design having b blocks, then there exists a c-FPC(v,b), where $c = \lfloor (k-1)/(t-1) \rfloor$.*

Similarly, we have the following construction for traceability schemes, using the same argument as in the proof of Theorem 3.3.

THEOREM 3.10. *If there exists a t-$(v,k,1)$ packing design having b blocks, then there exists a c-TS(k,b,v), where $c = \left\lfloor \sqrt{(k-1)/(t-1)} \right\rfloor$.*

We mentioned previously that no t-$(v,k,1)$ designs are known to exist if $v > k > t \geq 6$. However, for any t, there are infinite classes of packing designs with a "large" number of blocks (i.e., close to $\binom{v}{t}/\binom{k}{t}$). These can be obtained from designs known as orthogonal arrays, which are defined as follows.

DEFINITION 3.11. *An* orthogonal array *$OA(t,k,s)$ is a $k \times s^t$ array, with entries from a set of $s \geq 2$ symbols, such that in any t rows, every $t \times 1$ column vector appears exactly once.*

It is easy to obtain a packing from an orthogonal array, as shown in the next lemma.

LEMMA 3.12. *If there is an $OA(t,k,s)$, then there is a t-$(ks,k,1)$ packing design that contains s^t blocks.*

Proof. Suppose that there is a $\mathrm{OA}(t,k,s)$ with entries from the set $\{0,1,\ldots,s-1\}$. Define $X = \{(x,y) : 0 \leq x \leq k-1, 0 \leq y \leq s-1\}$. For every column $(y_0,y_1,\ldots,y_{k-1})$ in the orthogonal array, define a block $B = \{(0,y_0),(1,y_1),\ldots,(k-1,y_{k-1})\}$. Let $\mathcal{B}$ consist of the s^t blocks thus constructed. It is easy to check that $(X,\mathcal{B})$ is a t-$(ks,k,1)$ packing design. □

The following Lemma ([9, Chapter VI.7]) provides infinite classes of orthogonal arrays, for any integer t.

LEMMA 3.13. *If q is a prime power and $t < q$, then there exists an $OA(t,q+1,q)$, and hence a t-$\left(q^2+q, q+1, 1\right)$ packing design with q^t blocks exists.*

From Theorem 3.9 and Lemma 3.13, we obtain the following.

THEOREM 3.14. *For any prime power q and any integer $t < q$, there exists a $\left\lfloor \frac{q}{t-1} \right\rfloor$-FPC$(q^2+q, q^t)$ and a $\left\lfloor \sqrt{\frac{q}{t-1}} \right\rfloor$-TS$(q+1, q^t, q^2+q)$.*

In this construction, $b \approx 2^{\frac{\sqrt{v}\log v}{2c}}$ (for frameproof codes) and $b \approx 2^{\frac{\sqrt{v}\log v}{2c^2}}$ (for traceability schemes). Also, the resulting traceability schemes are of the "transversal type" considered in [8].

3.3. Constructions using perfect hash families. In this section, we present another method to construct frameproof codes, which uses a perfect hash family.

DEFINITION 3.15. *An (n,m,w)-*perfect hash family *is a set of functions $\mathcal{F}$ such that $f : \{1,2,\ldots,n\} \to \{1,2,\ldots,m\}$ for each $f \in \mathcal{F}$, and for any $X \subseteq \{1,2,\ldots,n\}$ such that $|X| = w$, there exists at least one $f \in \mathcal{F}$ such that $f|_X$ is one-to-one.*

When $|\mathcal{F}| = N$, an (n,m,w)-perfect hash family will be denoted by $\mathrm{PHF}(N;n,m,w)$. Observe that a $\mathrm{PHF}(N;n,m,w)$ can be depicted as an $N \times n$ matrix with entries from $\{1,2,\ldots,m\}$, having the property that in any w columns

there exists at least one row such that the w entries in the given w columns are distinct. Results on perfect hash families can be found in numerous textbooks and papers. Mehlhorn [12] is a good textbook source; more recent constructions can be found in the papers [2] and [3].

The following theorem tells us how to use a perfect hash family to enlarge a frameproof code.

THEOREM 3.16. *If there exists a PHF$(N; n, m, c+1)$ and a c-FPC(v, m), then there exists a c-FPC(Nv, n).*

Proof. Let $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(m)}\}$ be a c-FPC(v, m), and let $\mathcal{F}$ be a PHF$(N; n, m, c+1)$. Let Γ' be the (Nv, n)-code consisting of the n codewords

$$u^{(j)} = \biguplus_{h \in \mathcal{F}} w^{h(j)},$$

$j = 1, \ldots, n$, where $\uplus$ means concatenation of strings. We will show that Γ' is a c-FPC(Nv, n).

Let $W \subseteq \Gamma'$, $W = \{u^{(i_1)}, u^{(i_2)}, \ldots, u^{(i_c)}\}$. Recall that $U(W)$ is the set of undetectable bit positions of W. Assume that there exists a codeword $u^{(i_{c+1})} \in \Gamma' \backslash W$ such that $u^{(i_{c+1})}|_{U(W)} = u^{(i_j)}|_{U(W)}$ for $1 \le j \le c$. Since $\mathcal{F}$ is a PHF$(N; n, m, c+1)$, there exists an $h \in \mathcal{F}$ such that $h|_C$ is one-to-one, where $C = \{i_1, i_2, \ldots, i_{c+1}\}$. Thus we have $c+1$ different codewords $w^{(h(i_j))} \in \Gamma$, $1 \le j \le c+1$, such that $w^{(h(i_{c+1}))}$ is in the feasible set of $\{w^{(h(i_j))} : 1 \le j \le c\}$. This contradicts the fact that Γ is c-frameproof. □

In [4], the following construction of c-frameproof codes from error-correcting codes is given.

THEOREM 3.17. *If there exists a c-FPC(v, q) and an (N, n) q-ary code with minimum Hamming distance $d_{min} > N(1 - 1/c)$, then there exists a c-FPC(vN, n).*

Alon [1] gave a construction of perfect hash families from error-correcting codes. We observe that if we use a perfect hash family constructed by Alon's method to obtain a c-frameproof code by applying Theorem 3.16, then the resulting code is essentially the same as the one constructed using Theorem 3.17. However, it is possible to use other constructions for perfect hash families to obtain new examples of frameproof codes. We provide one illustration now, which uses the following recursive construction from [3].

LEMMA 3.18. *Suppose there exists a PHF$(N_0; n_0, m, w)$, where* $\gcd\left(n_0, \binom{w}{2}!\right) = 1$. *Then there exists a PHF $\left(\left(\binom{w}{2}+1\right)^j N_0; n_0^{2^j}, m, w\right)$ for any integer $j \ge 1$.*

Example 3.1. There exists a PHF$(2; 5, 4, 3)$ as follows:

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 3 \\ 1 & 1 & 2 & 1 & 3 \end{array} \qquad \square$$

THEOREM 3.19. *For any integer $j \ge 1$, there exists a 2-FPC$(6 \times 4^j, 5^{2^j})$.*

Proof. From Lemma 3.18 and Example 3.1, we obtain a PHF$(2 \times 4^j; 5^{2^j}, 4, 3)$ for all $j \ge 1$. Combine this perfect hash family with the 2-FPC$(3, 4)$ given in Example 1.2, and apply Theorem 3.16. □

4. Embeddings. In many cases the number of users of a scheme will increase after the system is set up. Initially, the data supplier will construct a scheme that will accommodate a fixed number of users (which we denoted by b). If the number of users eventually surpasses b, we would like a simple method of extending the scheme which

is "compatible" with the existing scheme. In the case of a traceability scheme, we do not want to change the personal keys already issued when the scheme is expanded. In the case of a frameproof code, we do not want to have to recall software that has already been sold.

To solve this problem, we will introduce the concept of embedding frameproof codes and traceability schemes in larger ones.

DEFINITION 4.1. *Let Γ be a c-FPC(v, b) and let Γ' be a c-FPC(v', b'), where $v < v'$, $b < b'$. Suppose that, for every codeword $w \in \Gamma$, there exists a codeword $w' \in \Gamma'$ such that the first v bit positions of w' are the same as w, and the remaining $v' - v$ bit positions of w' are all 0's. Then we say that Γ is* embedded *into Γ'.*

Initially, the distributor could use the code Γ to mark the products. When the number of users surpasses b, then codewords in $\Gamma' \backslash \Gamma$ are used. Note that the embedding property ensures that the codewords in Γ do not have to be changed when we proceed to the larger code.

A similar definition can be given for traceability schemes.

DEFINITION 4.2. *Let T be the set of v base keys of a c-TS(k, b, v), and let T' be the set of v' base keys of a c-TS(k, b', v'), where $v < v'$, $b < b'$ and $T \subseteq T'$. Suppose that every personal key of the c-TS(k, b, v) is also a personal key of the c-TS(k, b', v'). Then we say that the first scheme is* embedded *into the second scheme.*

Note that the definition of embedding is even simpler if we consider the set system formulation of frameproof codes and traceability schemes. Namely, we say that $(X, \mathcal{B})$ is *embedded* into $(X', \mathcal{B}')$ if $X \subseteq X'$ and $\mathcal{B} \subseteq \mathcal{B}'$.

Since t-designs and packing designs are set systems, the above definition of embedding applies. In fact, embeddings of combinatorial designs have been extensively studied, so we have a convenient method of constructing frameproof codes and traceability schemes which can be embedded.

For example, in the case of 2-designs, we have the following result.

THEOREM 4.3. *If there exists a 2-$(v, k, 1)$ design that can be embedded into a 2-$(v', k, 1)$ design, then there exists a $(k-1)$-FPC$(v, (v^2 - v)/(k^2 - k))$ that can be embedded into a $(k-1)$-FPC$(v', ((v')^2 - v')/(k^2 - k))$; and a $\lfloor \sqrt{k-1} \rfloor$-TS$(k, (v^2 - v)/(k^2 - k), v)$ that can be embedded into a $\lfloor \sqrt{k-1} \rfloor$-TS$(k, ((v')^2 - v')/(k^2 - k), v')$.*

We give a couple of illustrations of this idea. For $k = 3$ and 4, necessary and sufficient conditions for embedding 2-$(v, k, 1)$ designs into 2-$(v', k, 1)$ designs are known, namely $v \equiv 1$ or $k \bmod (k^2 - k)$, $v' \equiv 1$ or $k \bmod (k^2 - k)$, and $v' \geq (k-1)v + 1$. (For $k = 3$, this result is known as the Doyen-Wilson Theorem [9, Chapter I.4]; for $k = 4$, see [9, Chapter III.1].) This provides a convenient way of embedding 2- and 3-frameproof codes into larger ones by application of Theorem 4.3. The following theorems are obtained.

THEOREM 4.4. *For all $v \equiv 1, 3 \bmod 6$ and $v' \equiv 1, 3 \bmod 6$ such that $v' \geq 2v + 1$, there exists a 2-FPC$(v, (v^2 - v)/6)$ that can be embedded into a 2-FPC$(v', ((v')^2 - v')/6)$.*

THEOREM 4.5. *For all $v \equiv 1, 4 \bmod 12$ and $v' \equiv 1, 4 \bmod 12$ such that $v' \geq 3v+1$, there exists a 3-FPC$(v, (v^2 - v)/12)$ that can be embedded into a 2-FPC$(v', ((v')^2 - v')/12)$.*

Here is a small example to illustrate.

Example 4.1. Given an embedding of a 2-$(7, 3, 1)$ design into a 2-$(15, 3, 1)$ design, a 2-FPC$(7, 7)$ can be embedded into a 2-FPC$(15, 35)$. The 35 codewords of the 2-FPC$(15, 35)$ are given in Figure 4.1 (the first seven codewords, when restricted to the first seven bit positions, form the embedded 2-FPC$(7, 7)$). □

1101000	00000000
0110100	00000000
0011010	00000000
0001101	00000000
1000110	00000000
0100011	00000000
1010001	00000000
1000000	00110000
0100000	00011000
0010000	00001100
0001000	00000110
0000100	10000010
0000010	11000000
0000001	01100000
1000000	00001010
0100000	10000100
0010000	01000010
0001000	10100000
0000100	01010000
0000010	00101000
0000001	00010100
1000000	01000100
0100000	00100010
0010000	10010000
0001000	01001000
0000100	00100100
0000010	00010010
0000001	10001000
1000000	10000001
0100000	01000001
0010000	00100001
0001000	00010001
0000100	00001001
0000010	00000101
0000001	00000011

FIG. 4.1. *A* 2-*FPC*$(7,7)$ *embedded into a* 2-*FPC*$(15,35)$.

It is also well known that for any prime power q and for any integers $i \le j$, there exists a 2-$(q^i, q, 1)$ design which can be embedded into a 2-$(q^j, q, 1)$ design (in other words, the affine geometry $\mathrm{AG}(i,q)$ is a subgeometry of $\mathrm{AG}(j,q)$; see [9, Chapter VI.7]). The following result is obtained.

THEOREM 4.6. *Let q be a prime power, and let i and j be positive integers such that $i \le j$. Then there exists a $(q-1)$-FPC$(q^i, q^{i-1}(q^i-1)/(q-1))$ which can be embedded into a $(q-1)$-FPC$(q^j, q^{j-1}(q^j-1)/(q-1))$, and a $\lfloor\sqrt{q-1}\rfloor$-TS$(q, q^{i-1}(q^i-1)/(q-1), q^i)$ which can be embedded into a $\lfloor\sqrt{q-1}\rfloor$-TS$(q, q^{j-1}(q^j-1)/(q-1), q^j)$.*

5. Bounds. In this section, we investigate necessary conditions for existence for frameproof codes and traceability schemes. These take the form of upper bounds on b, as a function of c and v (and k, in the case of traceability schemes).

First we will give a bound for frameproof codes. Let $\Gamma = \{w^{(1)}, \ldots, w^{(b)}\}$ be

a c-FPC(v, b). Recall that $\mathcal{U}(C)$ denotes the set of undetectable bit positions for a subset $C \subseteq \Gamma$ and $F(C)$ denotes the feasible set of C. For $1 \leq d \leq c$, let

$$t_d = \min\{|\mathcal{U}(C)| : C \subseteq \{1, \ldots, b\}, |C| = d\}.$$

We begin by stating and proving a simple lemma.

LEMMA 5.1. *Suppose* $\Gamma = \{w^{(1)}, \ldots, w^{(b)}\}$ *is a* c*-FPC*(v, b)*, and suppose* $t_1, \ldots, t_c$ *are as defined above. Then* $0 < t_c < t_{c-1} < \ldots < t_1 = v$.

Proof. Suppose $t_d = t_{d-1}$ for some d. Let $C = \{w^{(u_1)}, \ldots, w^{(u_d)}\} \subseteq \Gamma$ be such that $|\mathcal{U}(C)| = t_d$, and let $C' = \{w^{(u_1)}, \ldots, w^{(u_{d-1})}\}$. Clearly $\mathcal{U}(C) \subseteq \mathcal{U}(C')$; however, since $t_d = t_{d-1}$, it follows that $\mathcal{U}(C) = \mathcal{U}(C')$. But then $C \subseteq F(C') \cap \Gamma$, which contradicts Definition 1.1. □

The next result provides an upper bound on b which depends on t_{c-1}.

THEOREM 5.2. *Suppose* $\Gamma = \{w^{(1)}, \ldots, w^{(b)}\}$ *is a* c*-FPC*(v, b)*, and suppose* $t_1, \ldots, t_c$ *are as defined above. Then*

$$b \leq c - 1 + \binom{t_{c-1}}{\lceil \frac{t_{c-1}}{2} \rceil}.$$

Proof. Let $W \subseteq \Gamma$ be chosen such that $|W| = c - 1$ and $|R| = t_{c-1}$, where $R = \mathcal{U}(W)$. For any codeword $w^{(i)} \in \Gamma \backslash W$, let $R_i = \mathcal{U}(W \cup \{w^{(i)}\})$. It is easy to see that $R_i \subseteq R$ for all $w^{(i)} \in \Gamma \backslash W$. Further, $R_i \not\subseteq R_j$ for all $w^{(i)}, w^{(j)} \in \Gamma \backslash W$, $i \neq j$ (for if $R_i \subseteq R_j$, say, then $w^{(j)} \in F(W \cup \{w^{(i)}\})$, which contradicts the fact that Γ is c-frameproof). In other words, the subsets R_i constructed above form a *Sperner family* with respect to the ground set R. By Sperner's Theorem (see, for example [11, Theorem 6.3]), we see that

$$|\Gamma \backslash W| \leq \binom{t_{c-1}}{\lceil \frac{t_{c-1}}{2} \rceil}.$$

Since $|\Gamma \backslash W| = b - c + 1$, the result follows. □

The following bound on b is an immediate corollary.

COROLLARY 5.3. *If* Γ *is a* c*-FPC*(v, b)*, then*

$$b \leq c - 1 + \binom{v - c + 2}{\lceil \frac{v-c+2}{2} \rceil}.$$

Proof. From Lemma 5.1, we have that $t_{c-1} \leq v - c + 2$. The conclusion follows from Theorem 5.2. □

Recall that Example 1.1 gave a construction for c-FPC(c, c), and we constructed a 2-FPC$(3, 4)$ in Example 1.2. In both of these examples, the bound of Corollary 5.3 is met with equality.

Now we turn our attention to traceability schemes, where we provide an upper bound on b. In [8], it was shown that $b \leq v^{k/c}$ if a c-TS(k, v, b) exists. We give a stronger bound, which is also based on the following observation made in [8].

LEMMA 5.4. *Suppose* $(X, \mathcal{B})$ *is a* c*-TS*(k, v, b)*. Then, for any subset of* $d \leq c$ *blocks* $B_1, B_2, \ldots, B_d \in \mathcal{B}$*, there does not exist a block* $B \in \mathcal{B} \backslash \{B_1, B_2, \ldots, B_d\}$ *such that* $B \subseteq \bigcup_{i=1}^{d} B_i$.

Proof. The proof is essentially the same as the proof of Theorem 2.3. □

For obvious reasons, the collection of subsets $\mathcal{B}$ is called *c-cover-free*. Now, applying [10, Proposition 2.1], which gives an upper bound on the cardinality of a c-cover-free collection of sets, the following result is immediate.

THEOREM 5.5. *If a c-$TS(k, b, v)$ exists, then the following bound holds:*

$$b \leq \frac{\binom{v}{t}}{\binom{k-1}{t-1}},$$

where $t = \lceil \frac{k}{c} \rceil$.

6. Comments. Further results on frameproof codes can be found in the Ph.D. thesis of Yeow Meng Chee [7, Chapter 9]. Chee gives a probabilistic construction for 2-frameproof codes that improves upon results in [4] and provides efficient explicit constructions for frameproof codes using the idea of superimposed codes.

Acknowledgments. We thank the referee for several helpful comments.

REFERENCES

[1] N. ALON, *Explicit construction of exponential sized family of k-independent sets*, Discrete Math., 58 (1986), pp. 191–193.
[2] N. ALON AND M. NAOR, *Derandomization, Witnesses for Boolean Matrix Multiplication and Constructions of Perfect Hash Functions*, Technical Report CS94-11, Weizmann Institute of Science, Rehovot, Israel.
[3] M. ATICI, S. S. MAGLIVERAS, D. R. STINSON, AND W.-D. WEI, *Some recursive constructions for perfect hash families*, J. Combin. Designs, 4 (1996), pp. 353–363.
[4] D. BONEH AND J. SHAW, *Collusion-secure fingerprinting for digital data*, in Advances in Cryptology – CRYPTO '95, D. Coppersmith, ed., Lecture Notes in Comput. Sci. 963, Springer-Verlag, Berlin, 1995, pp. 452–465.
[5] J. T. BRASSIL, S. LOW, N. F. MAXEMCHUK, AND L. O'GORMAN, *Electronic marking and identification techniques to discourage document copying*, IEEE J. Selected Areas in Communications, 13 (1995), pp. 1495–1503.
[6] M. CARAGIU, *On a class of constant weight codes*, Electronic J. Combin., 3 (1996), # R4, http://www.combinatorics.org.
[7] Y. M. CHEE, *Turán-type Problems in Group Testing, Coding Theory and Cryptography*, Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, 1996.
[8] B. CHOR, A. FIAT, AND M. NAOR, *Tracing traitors*, in Advances in Cryptology – CRYPTO '94, Y. G. Desmedt, ed., Lecture Notes in Computer Science, 839 (1994), pp. 257–270.
[9] C. J. COLBOURN AND J. H. DINITZ, EDS., *CRC Handbook of Combinatorial Designs*, CRC Press, Inc., Boca Raton, FL, 1996.
[10] P. ERDÖS, P. FRANKL, AND Z. FÜREDI, *Families of finite sets in which no set is covered by the union of r others*, Israel J. Math., 51 (1985), pp. 79–89.
[11] J. H. VAN LINT AND R. M. WILSON, *A Course in Combinatorics*, Cambridge University Press, Cambridge, 1992.
[12] K. MEHLHORN, *Data Structures and Algorithms*, Vol. 1., Springer-Verlag, New York, 1984.
[13] A. SHAMIR, *How to share a secret*, Comm. ACM, 22 (1979), pp. 612–613.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 54–60, February 1998

004

THE NUMBER OF INDEPENDENT SETS IN A GRID GRAPH*

NEIL J. CALKIN† AND HERBERT S. WILF‡

Abstract. If $f(m,n)$ is the (vertex) independence number of the $m \times n$ grid graph, then we show that the double limit $\eta \stackrel{\text{def}}{=} \lim_{m,n\to\infty} f(m,n)^{\frac{1}{mn}}$ exists, thereby refining earlier results of Weber [*Rostock. Math. Kolloq.*, 34 (1988), pp. 28–36] and Engel [*Fibonacci Quart.*, 28 (1990), pp. 72–78]. We establish upper and lower bounds for η and *prove* that $1.503047782\ldots \le \eta \le 1.5035148\ldots$. Numerical computations suggest that the true value of η (the "hard square constant") is around 1.5030480824753323 ....

Key words. independent sets, grid graphs

AMS subject classifications. 05A16, 05C50, 82B20

PII. S089548019528993X

Let $G_{m,n}$ be the $m \times n$ grid graph. That is, the vertices of $G_{m,n}$ are the $(m+1)(n+1)$ points (i,j) $(0 \le i \le m, 0 \le j \le n)$ in the plane, and its edges consist of the pairs $(i,j),(i',j')$ of vertices for which $|i-i'|+|j-j'|=1$. Let $f(m,n)$ be the number of independent sets of vertices in $G_{m,n}$. We study the growth of $f(m,n)$. Figure 1 shows an independent set S in $G_{4,6}$.

Clearly $G_{m,n}$ contains an independent set of size $\ge mn/2$, and that set has $\ge 2^{mn/2}$ subsets, so certainly

$$\liminf_{m,n\to\infty} f(m,n)^{\frac{1}{mn}} \ge \sqrt{2} = 1.4142\ldots.$$

In [2], Weber showed the existence of the limits

$$\lim_{n\to\infty} f(m,n)^{1/mn} \qquad \text{and} \qquad \lim_{n\to\infty} f(n,n)^{1/n^2},$$

and estimated their values. In [1], Engel proved some inequalities for these quantities, deduced that $1.50304808 \le \eta \le 1.51316067$, and conjectured that $\eta = 1.50304808\ldots$.

We will prove that the double limit

$$\eta \stackrel{\text{def}}{=} \lim_{m,n\to\infty} f(m,n)^{\frac{1}{mn}} \tag{1}$$

exists, and that $1.503047782\ldots \le \eta \le 1.5035148\ldots$, the latter by exhibiting (relatively) easily computable upper and lower bounds. Numerical computations suggest that the true value of η is around 1.5030480824753323 ....

1. The transfer matrix. We use the transfer matrix method, in a manner that is similar to the way it was used in [3]. Let S be an independent set in $G_{m,n}$. Consider the portion of S that lies in a fixed column of the graph. This can be regarded as an $(m+1)$-vector of 0's and 1's, in which a 1 indicates that the vertex is in S, and a 0

*Received by the editors August 3, 1995; accepted for publication (in revised form) November 12, 1996. This research was supported in part by the Office of Naval Research.
http://www.siam.org/journals/sidma/11-1/28993.html
†School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 (calkin@math.gatech.edu).
‡Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104-6395 (wilf@central.cis.upenn.edu).

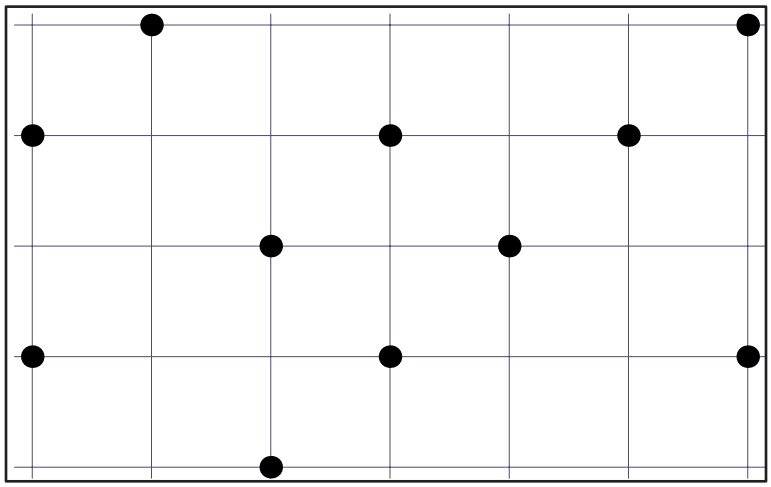


FIG. 1.

indicates that the vertex is not in S. The $(m+1)$-vectors that can arise this way are those that have the property that no two consecutive 1's occur.

Example. Figure 1 above shows an independent set S in $G_{4,6}$. The portions of S that lie in each of the seven columns can be represented by the respective 5-vectors

$$(0,1,0,1,0),(1,0,0,0,0),(0,0,1,0,1),(0,1,0,1,0),(0,0,1,0,0),(0,1,0,0,0),(1,0,0,1,0).$$

Each of these 5-vectors has the property that no two consecutive 1's occur. □

Thus, for m, n fixed, we can think of assembling the independent sets of the grid graph by gluing together columns that are chosen from the collection of possible columns, making sure that when we glue an additional column onto the right-hand edge of the structure, the new column does not clash with the previous right-hand column.

The collection of possible columns $\mathcal{C}_m$ is the set of all $(m+1)$-vectors $\mathbf{v}$, of 0's and 1's, such that $\mathbf{v}$ contains no two consecutive 1's. The number of these is well known to be F_{m+2}, the Fibonacci number.

The condition that vectors $\mathbf{v}'$, $\mathbf{v}''$ in $\mathcal{C}_m$ are a possible consecutive pair of columns in an independent set of $G_{m,n}$ is simply that they have no 1's in common position, i.e., that $\mathbf{v}' \cdot \mathbf{v}'' = 0$ in the sense of the usual dot product of vectors over the reals.

Thus, all possible independent sets in the grid graph are obtained by beginning with some vector of $\mathcal{C}_m$, and in general, having arrived at some sequence of vectors of $\mathcal{C}_m$, adjoin any vector of $\mathcal{C}_m$ that is orthogonal to the last one previously chosen until $(n+1)$ vectors have been selected.

We define a matrix $T = T_m$, the transfer matrix of the problem, as follows. T is an $F_{m+2} \times F_{m+2}$ symmetric matrix of 0's and 1's whose rows and columns are indexed by vectors of $\mathcal{C}_m$. The entry of T in position $(\mathbf{v}', \mathbf{v}'')$ is 1 if the vectors $\mathbf{v}', \mathbf{v}''$ are orthogonal, and is 0 otherwise. T depends only on m, not on n.

Let $f(m, n, \mathbf{u})$ denote the number of independent sets of $G_{m,n}$ whose rightmost column vector is $\mathbf{u}$. Then, clearly, we have

$$f(m, n+1, \mathbf{v}) = \sum_{\mathbf{u} \in \mathcal{C}_m} f(m, n, \mathbf{u}) T_{\mathbf{u},\mathbf{v}} \qquad (n \geq 0; \mathbf{v} \in \mathcal{C}_m),$$

or, in matrix-vector notation, $\mathbf{f}_{n+1} = T\mathbf{f}_n$, with $\mathbf{f}_0 = \mathbf{1}$ the vector whose entries are all 1's. It follows that $\mathbf{f}_n = T^n\mathbf{1}$, for all $n \geq 0$. The number of independent sets of $G_{m,n}$ is the sum of the entries of the vector $\mathbf{f}_n$. Thus

$$f(m,n) = \mathbf{1} \cdot T^n \mathbf{1},$$

i.e., $f(m,n)$ is the sum of all of the entries of the matrix T^n.

Since T has nonnegative entries, its dominant eigenvector cannot be orthogonal to $\mathbf{1}$, and so we have at once that $\lim_{n\to\infty} f(m,n)^{1/n}$ exists for each m, and is equal to Λ_m, the largest eigenvalue of the (real, symmetric) transfer matrix T (the existence of that limit also follows from an obvious subadditivity argument). It follows that

$$\liminf \Lambda_m^{1/m} = \liminf_{m,n} f(m,n)^{1/mn} \leq \limsup_{m,n} f(m,n)^{1/mn} = \limsup \Lambda_m^{1/m}.$$

We remark in passing that some interesting generating functions can be found in moderately explicit form. Indeed, since $f(m,n)$ is the sum of the entries of T^n, we see that for m fixed, the numbers $f(m,n)$ can be read off as the coefficient of x^n in the power series expansion of the sum of the entries of the matrix $(I - xT)^{-1}$.

For instance, take $m = 2$. The possible column vectors in an independent set are

$$(000), (001), (010), (100), (101).$$

If we index the rows and columns in this order, then the transfer matrix is

$$T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{2}$$

If we find the sum of the entries of $(I - xT)^{-1}$ we see that $f(2,n)$ is the coefficient of x^n in

$$\frac{5 + 7\,x - x^2 - x^3}{1 - 2\,x - 6\,x^2 + x^4} = 5 + 17\,x + 63\,x^2 + 227\,x^3 + 827\,x^4 + 2999\,x^5 + 10897\,x^6 + O(x^7).$$

2. The lower bound. In this section we will first prove the existence of the limit η, and then by a slight refinement of the argument, we will give a close lower bound for η.

By the maximum principle, since the transfer matrix T is real and symmetric, we have, for every positive integer p,

$$\Lambda_m^p \geq \frac{(\mathbf{1}, T_m^p \mathbf{1})}{(\mathbf{1}, \mathbf{1})}, \tag{3}$$

where we now explicitly exhibit the dependence of the transfer matrix on m by the subscript. But $(\mathbf{1}, T_m^p \mathbf{1}) = (\mathbf{1}, T_p^m \mathbf{1})$, since both sides count the independent sets in the grid graph $G_{m,p}$. Thus, after taking mth roots we have

$$(\Lambda_m^{1/m})^p \geq \left(\frac{(\mathbf{1}, T_p^m \mathbf{1})}{(\mathbf{1}, \mathbf{1})} \right)^{1/m}. \tag{4}$$

Now take the lim inf of both sides of this inequality, as $m \to \infty$. We obtain

$$(\liminf_{m\to\infty} \Lambda_m^{1/m})^p \geq \frac{\Lambda_p}{\frac{1+\sqrt{5}}{2}}. \tag{5}$$

Now take the pth root, and the lim sup as $p \to \infty$ to discover that

$$\liminf_{m\to\infty} \Lambda_m^{1/m} \geq \limsup_{p\to\infty} \Lambda_p^{1/p}.$$

The reverse inequality being obvious, we have that the limit $\lim_{m\to\infty} \Lambda_m^{1/m}$ exists, and hence by (3) so does the limit

$$\eta = \lim_{m,n} f(m,n)^{1/mn} = \lim_{m\to\infty} \Lambda_m^{1/m}. \qquad \square \tag{6}$$

Next we will refine the above argument to obtain a good numerical lower bound for η. We replace (4) by

$$\Lambda_m^p \geq \frac{(T_m^q \mathbf{1}, T_m^p T_m^q \mathbf{1})}{(T_m^q \mathbf{1}, T_m^q \mathbf{1})},$$

which, by the maximum principle, is true for every positive integer q. But the right-hand side can be rewritten as

$$\frac{(\mathbf{1}, T_m^q T_m^p T_m^q \mathbf{1})}{(\mathbf{1}, T_m^q T_m^q \mathbf{1})} = \frac{(\mathbf{1}, T_m^{p+2q} \mathbf{1})}{(\mathbf{1}, T_m^{2q} \mathbf{1})} = \frac{(\mathbf{1}, T_{p+2q}^m \mathbf{1})}{(\mathbf{1}, T_{2q}^m \mathbf{1})},$$

where we have again used the fact that $\forall p, m : T_m^p = T_p^m$. Hence

$$\eta^p = \lim_{m\to\infty} (\Lambda_m^{1/m})^p \geq \frac{\Lambda_{p+2q}}{\Lambda_{2q}},$$

and so

$$\eta \geq \left(\frac{\Lambda_{p+2q}}{\Lambda_{2q}} \right)^{\frac{1}{p}}. \tag{7}$$

Example. We now work out the case $p = 2$, $q = 1$ of this lower bound. The transfer matrix T_2 is shown in (2) above, and its largest eigenvalue is the largest zero of $1 - 6\,x^2 - 2\,x^3 + x^4 = 0$, i.e., $\Lambda_2 = 3.6313812604036\ldots$.

The transfer matrix T_4 is 13×13, and it is given by

$$T_4 = \begin{array}{c} \\ \emptyset \\ v \\ w \\ x \\ y \\ z \\ v\wedge x \\ v\wedge y \\ v\wedge z \\ w\wedge y \\ w\wedge z \\ x\wedge z \\ v\wedge x\wedge z \end{array} \begin{array}{c} \begin{array}{ccccccccccccc} \emptyset & v & w & x & y & z & v\wedge x & v\wedge y & v\wedge z & w\wedge y & w\wedge z & x\wedge z & v\wedge x\wedge z \end{array} \\ \left(\begin{array}{ccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right) \end{array}.$$

The largest eigenvalue of T_4 is the largest root of the equation

$$1 - 4t - 20t^2 + 64t^3 + 15t^4 - 105t^5 + 36t^6 + 4t^7 - t^8 = 0,$$

namely, $\Lambda_4 = 8.2032591937550246879103\ldots$. Hence we have the lower bound

$$\eta \geq \left(\frac{\Lambda_4}{\Lambda_2}\right)^{\frac{1}{2}} = 1.502994159\ldots. \qquad \square$$

We have in fact worked out the case $p = 2$, $q = 3$, though we will not show the details here, with the result that $\eta \geq 1.503047782\ldots$.

3. The upper bound. In this section we will exhibit another transfer matrix problem with the property that it provides upper bounds for the problem in which we are interested. Further, the upper bounding problem will be independent of m, n, and will depend on a new integer parameter p. There will be a valid upper bound for each positive integer p.

For each positive integer p, the largest eigenvalue of the transfer matrix T obviously satisfies

$$\Lambda_m \leq \text{Trace}(T^{2p})^{1/2p}, \tag{8}$$

and indeed the right-hand side approaches the left for $p \to \infty$. However,

$$\text{Trace}(T^{2p}) = \sum T_{x_0,x_1} T_{x_1,x_2} \cdots T_{x_{2p-1},x_0}.$$

Now each term in this sum is 0 or 1; hence the sum is equal to the number of *good* $2p$*-tuples* of subsets of $1, 2, \ldots, m$, that is, the number of $2p$-tuples $(x_0, x_1, \ldots, x_{2p-1})$ of subsets of $1, 2, \ldots, m$ for which

(a) each x_i contains no two consecutive entries, and

(b) each consecutive (on the circle) pair of x's is disjoint.

We will find another way to count these tuples that will enable us to eliminate the dependence on m completely.

Define the associated transfer matrix B_{2p} to be the matrix whose rows and columns are indexed by all subsets of $2p$ abstract "objects" which contain no two objects that are consecutive (on the circle). The matrix is $N_p \times N_p$, where $N_p = F_{2p-1} + F_{2p+1}$, and the F's are the Fibonacci numbers. The entries of this matrix are

$$(B_{2p})_{X,Y} = \begin{cases} 1, & \text{if } X \cap Y = \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

Note that this matrix is independent of m. Its utility rests in the following fact.

PROPOSITION. *The trace of T^{2p} is equal to the sum of all of the entries of the matrix B_{2p}^{m-1}.*

Proof. Consider a nonvanishing term of $\text{Trace}(T^{2p})$, say, the term $T_{x_0,x_1} T_{x_1,x_2} \cdots T_{x_{2p-1},x_0}$. Now each x_i is a subset of $1, 2, \ldots, m$. Define sets

$$S_j = \{k : (0 \leq k \leq 2p-1) \wedge (j \in x_k)\} \qquad (j = 1, 2, \ldots, m).$$

Then each S_j is a subset of $2p$ objects. Each S_j contains no two objects that are consecutive on the circle, for otherwise j would belong to two consecutive x_k's on the circle. Further, S_i, S_{i+1} (on the circle) are disjoint, for otherwise two consecutive letters $i, i+1$ would both belong to one of the x_k's, which is a contradiction.

Hence this collection of sets S_j corresponds to a nonvanishing expression

$$(B_{2p})_{S_1,S_2}(B_{2p})_{S_2,S_3}\cdots(B_{2p})_{S_{m-1},S_m}. \tag{9}$$

But this is just one of the terms in the expansion of the sum of all of the entries of the matrix B_{2p}^{m-1}, i.e., in the expansion of $(\mathbf{1}, B_{2p}^{m-1}\mathbf{1})$.

Conversely, consider a nonvanishing term of $(\mathbf{1}, B_{2p}^{m-1}\mathbf{1})$, say, the term shown in (9) above. Define a $(2p)$-tuple $(x_0,\ldots,x_{2p-1})$ of subsets of $1,2,\ldots,m$ by

$$x_j = \big\{i : (1\le i\le m)\wedge(j\in S_i)\big\} \qquad (j=0,1,\ldots,2p-1).$$

Then each x_i has no two consecutive entries, for otherwise j would belong to two consecutive S_i's and one of the factors $(B_{2p})_{S_i,S_{i+1}}$ would vanish. Likewise, each consecutive (on the circle) pair of sets x_j is disjoint, for otherwise some S_i would contain two consecutive (on the circle) values of j. This completes the proof of the proposition. □

Now, for each fixed positive integer p we have

$$\Lambda_m \le \mathrm{Trace}(T^{2p})^{1/2p} = (\mathbf{1}, B_{2p}^{m-1}\mathbf{1}).$$

If we take the mth root and then the limit as $m\to\infty$, we find that

$$\eta = \lim_{m,n\to\infty} f(m,n)^{1/mn} = \limsup_{m\to\infty} \Lambda_m^{1/m} \le \xi_{2p}^{1/2p},$$

where ξ_{2p} is the largest eigenvalue of B_{2p}.

Example. We consider the case $2p=6$.

Here, $\mathrm{Trace}(T^6)$ is the number of good 6-tuples (u,v,w,x,y,z), such that each of the six is a subset of $[1,m]$, none of them contains any two consecutive elements, and all of the pairs

$$(u,v),\ (v,w),\ (w,x),\ (x,y),\ (y,z),\ (z,u)$$

are disjoint pairs. Thus, a single letter i might belong to any of the 18 combinations

$$\emptyset,\ u,\ v,\ w,\ x,\ y,\ z,\ u\wedge w,\ u\wedge x,\ u\wedge y,\ v\wedge x,\ v\wedge y,\ v\wedge z,\ w\wedge y,\ w\wedge z,\ x\wedge z,\ u\wedge w\wedge y,\ v\wedge x\wedge z.$$

The associated transfer matrix B_6 is 18×18, and its entries are 1 or 0 depending on whether the membership combination that is labeled by the row is disjoint from the membership combination that is labeled by the column. The full matrix B_6, with its

lines labelled in the order shown above, is

$$\begin{pmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&1&1&1&1&1&1&0&0&1&1&1&1&1&1&0&1\\
1&1&0&1&1&1&1&1&1&1&0&0&0&1&1&1&1&0\\
1&1&1&0&1&1&1&0&1&1&1&1&1&0&0&1&0&1\\
1&1&1&1&0&1&1&1&0&1&0&1&1&1&1&0&1&0\\
1&1&1&1&1&0&1&1&1&0&1&0&1&0&1&1&0&1\\
1&1&1&1&1&1&0&1&1&1&1&1&0&1&0&0&1&0\\
1&0&1&0&1&1&1&0&0&0&1&1&1&0&0&1&0&1\\
1&0&1&1&0&1&1&0&0&0&0&1&1&1&1&0&0&0\\
1&0&1&1&1&0&1&0&0&0&1&0&1&0&1&1&0&1\\
1&1&0&1&0&1&1&1&0&1&0&0&0&1&1&1&1&0\\
1&1&0&1&1&0&1&1&1&0&0&0&0&0&1&1&0&0\\
1&1&0&1&1&1&0&1&1&1&0&0&0&1&0&0&1&0\\
1&1&1&0&1&0&1&0&1&0&1&0&1&0&0&1&0&1\\
1&1&1&0&1&1&0&0&1&1&1&1&0&0&0&0&0&0\\
1&1&1&1&0&1&0&1&0&1&0&1&0&1&0&0&1&0\\
1&0&1&0&1&0&1&0&0&0&1&0&1&0&0&1&0&1\\
1&1&0&1&0&1&0&1&1&1&0&0&0&1&0&0&1&0
\end{pmatrix}.$$

Calculation then reveals that the largest eigenvalue of B_6 is the largest root of the equation

$$-1+2t+25t^2+3t^3-12t^4+t^5=0,$$

namely, $11.55170956604814509\ldots$. Therefore,

$$\eta=\lim_{m,n} f(m,n)^{1/mn}\le 11.55170956604814509\ldots^{1/6}=1.503514809475903023\ldots. \quad \square$$

REFERENCES

[1] Konrad Engel, *On the Fibonacci number of an $m\times n$ lattice*, Fibonacci Quart., 28 (1990), pp. 72–78.

[2] Karl Weber, *On the number of stable sets in an $m\times n$ lattice*, Rostock. Math. Kolloq., 34 (1988), pp. 28–36.

[3] Herbert S. Wilf, *The problem of the kings*, Electron. J. Combin., 2 (1995), #R3, http://www.combinatorics.org.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 61–80, February 1998

005

A RANDOMNESS-ROUNDS TRADEOFF IN PRIVATE COMPUTATION*

EYAL KUSHILEVITZ† AND ADI ROSÉN‡

Abstract. We study the role of randomness in multiparty private computations. In particular, we give several results that prove the existence of a randomness-rounds tradeoff in multiparty private computation of `xor`. We show that with a single random bit, $\Theta(n)$ rounds are necessary and sufficient to privately compute `xor` of n input bits. With $d \geq 2$ random bits, $\Omega(\log n/d)$ rounds are necessary, and $O(\log n/\log d)$ are sufficient.

More generally, we show that the private computation of a boolean function f, using $d \geq 2$ random bits, requires $\Omega(\log S(f)/d)$ rounds, where $S(f)$ is the sensitivity of f. Using a single random bit, $\Omega(S(f))$ rounds are necessary.

Key words. private distributed computations, randomness, lower bounds, sensitivity

AMS subject classifications. 68Q22, 68R05, 94A60

PII. S089548019427634X

1. Introduction.

A 1-*private* (or simply, *private*) protocol $\mathcal{A}$ for computing a function f is a protocol that allows n players, P_i, $1 \leq i \leq n$, each possessing an individual secret input, x_i, to compute the value of $f(\vec{x})$ in a way that no *single* player learns more about the initial inputs of other players than what is revealed by the value of $f(\vec{x})$ and its own input[1]. The players are assumed to be honest but curious. Namely, they all follow the prescribed protocol $\mathcal{A}$ but they could try to get additional information by considering the messages they receive during the execution of the protocol. Private computations in this setting were the subject of a considerable amount of work, e.g., [BGW88, CCD88, BB89, CK89, K89, B89, FY92, CK92, CGK90, CGK92, KMO94]. One crucial ingredient in private protocols is the use of *randomness*. Quantifying the amount of randomness needed for computing functions privately is the subject of the present work.

Randomness as a resource was extensively studied in the last decade. Methods for saving random bits range over pseudorandom generators [BM84, Y82, N90], techniques for recycling random bits [IZ89, CW89], sources of weak randomness [CG88, VV85, Z91], and constructions of different kinds of small probability spaces [NN90, AGHP90, S92, KM93, KM94, KK94] (which sometimes even allow one to eliminate the use of randomness). A different direction of research is a quantitative study of the role of randomness in specific contexts, e.g., [RS89, KPU88, BGG90,

* Received by the editors October 31, 1994; accepted for publication (in revised form) October 11, 1996. A preliminary version of this paper appeared in *Advances in Cryptology*, Proceedings of Crypto '94, Lecture Notes in Computer Science 839, Y. Desmedt, ed., Springer-Verlag, Berlin, 1994, pp. 397–410.

http://www.siam.org/journals/sidma/11-1/27634.html

† Department of Computer Science, Technion, Haifa 32000, Israel (eyalk@cs.technion.ac.il, http://www.cs.technion.ac.il/∼eyalk). The research of this author was supported by the E. and J. Bishop Research Fund and by the Fund for the Promotion of Research at the Technion. Part of this research was performed while the author was at Aiken Computation Laboratory, Harvard University, Cambridge, MA, and was supported by research contracts ONR-N0001491-J-1981 and NSF-CCR-90-07677.

‡ Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (adiro@math.tau.ac.il).

[1] In the literature a more general definition of t-privacy is given. The above definition is the case $t = 1$.

CG90, BGS94, BSV94]. In this work, we initiate a quantitative study of randomness in private computations. We mainly concentrate on the specific task of computing the `xor` of n input bits. However, most of our results extend to any boolean function. The task of computing `xor` was the subject of previous research due to its being a basic linear operation and to its relative simplicity [FY92, CK92].

It is known as a "folklore theorem" (and is not difficult to show) that private computation of `xor` cannot be carried out deterministically (for $n \geq 3$). On the other hand, with a single random bit, such a computation becomes possible: at the first round, player P_n chooses a random bit r and sends to P_1 the bit $x_n \oplus r$. Then, in round i ($2 \leq i \leq n$) player P_{i-1} xors its bit x_{i-1} with the message it received in the previous round and sends the result to P_i. Finally, P_n xors the message it received with the random bit r. Both the correctness and privacy of this protocol are easy to verify. The main drawback of this protocol is that it takes n rounds. Another protocol for this task computes `xor` in two rounds but requires a linear number of random bits: In the first round each player P_i chooses a random bit r_i. Then, player P_i sends $x_i \oplus r_i$ to P_1 and r_i to P_2. In the second round, P_2 xors all the (random) bits it received in the first round and sends the result to P_1 which xors all the messages it received during the protocol to get the value of the function. Again, both the correctness and privacy of this protocol are not hard to verify.

In this work we prove that there is a tradeoff between the amount of randomness and the number of rounds in private computations of `xor`. For example, we show that while with a single random bit $\Theta(n)$ rounds are necessary and sufficient[2], with two random bits $O(\log n)$ rounds suffice.[3] Namely, with a single additional random bit, the number of rounds is significantly reduced. Additional bits give a much more "modest" saving. More precisely, we prove that with $d \geq 2$ random bits, $O(\log n / \log d)$ rounds suffice and $\Omega(\log n/d)$ rounds are required. Our upper bound is achieved using a new method that enables us to use linear combinations of random bits again and again (while preserving the privacy). The lower bounds are proved using combinatorial arguments, and they are strong in the sense that they also apply to protocols that are allowed to make errors, and that they actually show a lower bound on the *expected* number of rounds. We also show that if protocols are restricted to certain natural types (that include, in particular, the protocol that achieves the upper bound) we can even improve the lower bound and show that $\Theta(\log n / \log d)$ rounds are necessary and sufficient.

Our lower bound techniques apply not only to the `xor` function but in fact give lower bounds on the number of rounds for any boolean function in terms of the sensitivity of the function. Namely, we prove that with $d \geq 2$ random bits, $\Omega(\log S(f)/d)$ rounds are necessary to privately compute a boolean function f, whose sensitivity is $S(f)$. With a single random bit ($d = 1$), $\Omega(S(f))$ rounds are necessary.

The question of whether private computations in general can be carried out in constant number of rounds was previously addressed [BB89, BFKR90]. In light of

[2] More precisely, $\lceil n/2 \rceil$ rounds. This upper bound is achieved by a slight modification of the first protocol above. Assume, for simplicity, that n is even. At the first round, player P_n sends $x_n \oplus r$ to player P_1, and at the same time sends r to player P_{n-1}. The players then continue as in the above protocol, forwarding messages in parallel until the two messages meet. More precisely, in round i, $2 \leq i \leq \frac{n}{2}$, player P_{i-1} xors the message it received with its own input and sends it to player P_i and player $P_{n-(i-1)}$ xors the message it received with its own input and sends it to player P_{n-i}. In round $\frac{n}{2}$, player $P_{n/2}$ receives two messages and can compute the value of the function by xoring the two messages together with its own input.

[3] All logarithms are base 2, unless otherwise indicated.

our results, a promising approach to investigate this question may be by proving that if a constant number of rounds is sufficient, then a large number of random bits is required.

Subsequent to our work, several other works were done regarding the amount of randomness in privacy. In particular, the amount of randomness required for computing the function `xor` t-privately, for $t \geq 2$, was studied in [BDPV95, KM96]; in [KOR96] it is shown that the boolean functions that can be computed privately with a constant number of random bits are exactly the functions that have linear-size boolean circuits. Further results on randomness in private computations appear in [CKOR97].

The rest of the paper is organized as follows: in section 2 we give some definitions. In section 3 we give an upper bound on the number of rounds required to privately compute `xor`. In section 4 we give lower bounds on the number of rounds to privately compute a boolean function, in terms of its sensitivity. In section 5 we give lower bounds on the expected number of rounds in terms of the average sensitivity of the function being computed. Section 6 includes some conclusions and ideas for further research. The appendix contains the improved lower bounds for restricted types of protocols.

2. Preliminaries. We give here a description of the protocols we consider, and define the *privacy* property of protocols. More rigorous definitions of the protocols are given in section 4.1.

Let $f : \{0,1\}^n \to \{0,1\}$ be any boolean function. A set of n players P_i $(1 \leq i \leq n)$, each possessing a single input bit x_i (known *only* to player P), collaborate in a protocol to compute the value of $f(\vec{x})$. The protocol operates in rounds. In each round each player may toss some coins, and then sends messages to the other players (messages are sent over private channels so that, other than the intended receiver, no other player can listen to them). It then receives the messages sent to it by the other players. In addition, each player at a certain round chooses to output the value of the function. We assume that each player knows its serial number and the total number of players n. We may also assume that each player P_i is provided with a read-only random tape R_i from which it reads random bits (rather than tossing coins).

During the execution of the protocol, each player P_i receives a sequence of messages. Let c_i be a random variable of the communication string seen by player P_i, and let C_i be a particular communication string seen by P_i. Informally, *privacy* with respect to player P_i means that player P_i cannot learn anything (in particular, the inputs of the other players) from C_i, except what is implied by its input bit, and the value of the function computed. Formally, we have the following definition.

DEFINITION 2.1 (privacy). *A protocol $\mathcal{A}$ for computing a function f is* private *with respect to player P_i if for any two input vectors $\vec{x}$ and $\vec{y}$, such that $f(\vec{x}) = f(\vec{y})$ and $x_i = y_i$, for any sequence of messages C_i, and for any random tape R_i provided to P_i,*

$$Pr[c_i = C_i | R_i, \vec{x}] = Pr[c_i = C_i | R_i, \vec{y}],$$

where the probability is over the random tapes of all other *players.*

3. Upper bound. This section presents a protocol which allows n players to use $d \geq 2$ random bits for computing `xor` privately. This protocol takes $O(\log_d n)$ rounds. (For the case $d = 1$ a similar protocol that uses $\lceil n/2 \rceil$ rounds was already described in the introduction.) All arithmetic operations in this section are done modulo 2.

Consider the following protocol (which we call the *basic protocol*): first organize the n players in a tree. The degree of the root of the tree is $d + 1$, and the degree of

any other internal node is d (assume for simplicity that n is such that this forms a complete tree). The computation starts from the leaves and goes towards the root by sending messages (each of them of a single bit) as follows: each leaf player P_i sends its input bit x_i to its parent in the tree. Each internal node, after receiving messages from its d children, sums them up (modulo 2), together with its input bit x_i, and sends the result to its parent. Finally, the root player sums up the $d+1$ messages it receives, together with its input bit, and the result is the output of the protocol.

While a simple induction shows the correctness of this protocol, and it clearly runs in $O(\log_d n)$ rounds, it is obvious that it does not maintain the required privacy. The second idea will be to "mask" each of the messages sent in the basic protocol by an appropriate random bit (constructed using the d-random bits available) in a way that these masks will disappear at the end, and we will be left with the (unmasked) output. To do so we assign the nodes of the above tree vectors in $GF[2^d]$ as follows (the meaning of those vectors will become clear soon): assign to the root the vector $(0,\ldots,0)$. The children of the root will be assigned $d+1$ (nonzero) vectors such that the vectors in any d-size subset of them are linearly independent and the sum of all the $d+1$ vectors is $(0,\ldots,0)$ (for example, the d unit vectors, together with the $(1,\ldots,1)$ vector, satisfy these requirements). Finally, in a recursive way, given an internal node which is assigned a vector v, we assign to its d children d linearly independent vectors whose sum is v (note that, in particular, none of these vectors is the $\vec{0}$ vector)[4].

We now show how to use the vectors we assigned to the nodes so as to get a private protocol. We will assume that the random bits $b_1,\ldots,b_d$ are chosen by some external processor. We will later see that this assumption can be eliminated easily. Let v be the vector assigned to some player which is a *leaf* in the tree. We will give this player a single bit $r_v = v \cdot b$, where $b = (b_1,\ldots,b_d)$ is the vector consisting of the d random bits and the product is an inner product (modulo 2). The players will use the basic protocol, described above, with the modification that a player in a leaf also xors its message with the bit r_v it received (the other players behave exactly as before). We claim that for every player P_i, if in the basic protocol it sends the message m when the input vector is $\vec{x}$, then in the modified protocol it sends the message $m+(v_i \cdot b)$, where v_i is the vector assigned to this player. The proof goes by induction: it is trivially true for the leaf players. For internal nodes the message is calculated by adding the input of the players to the sum of the incoming messages. Using the induction hypothesis this sum is $\sum_{k=1}^{d}[m^k + (v^k \cdot b)]$, where m^k is the message received from the kth child in the basic protocol, and v^k is the vector assigned to the kth child. Since the construction is such that v_i, the vector assigned to P_i, satisfies $v_i = \sum_{k=1}^{d} v^k$, then a simple algebraic manipulation proves the induction step. In particular, since the root is assigned the vector $(0,\ldots,0)$, its output equals the output of the basic protocol. Hence, the correctness follows.

We now prove the privacy property of the protocol. The leaf players do not receive any message, hence there is nothing to prove. Let P_j be an internal node in the tree. Denote by $s^1,\ldots,s^d$ the messages it receives. We claim that for every vector $w = (w_1,\ldots,w_d) \in GF[2^d]$, and for any input vector, we have

$$Pr[(s^1 = w_1) \wedge \cdots \wedge (s^d = w_d)] \;=\; \frac{1}{2^d},$$

[4] For example, such a collection of d vectors can be constructed as follows: since $v \neq \vec{0}$, there exists an index i such that $v_i = 1$. The first $d-1$ vectors will be the $d-1$ unit vectors $e_1,\ldots,e_{i-1},e_{i+1},\ldots,e_d$. The last vector will be $v - \sum_{j\neq i} e_j$. Obviously the sum of these d vectors is v and they are linearly independent.

where the probability is over the random choice of $b_1, \ldots, b_d$ (note that in this protocol the players do not make internal random choices). In other words, fix any specific input vector $\vec{x}$, then for every vector w there exists exactly one choice of values for $b_1, \ldots, b_d$ such that the (vector of) messages that P_j receives, when the protocol is executed with input $\vec{x}$, is identical to the vector w. Denote by $\vec{v}^1, \ldots, \vec{v}^d$ the vectors corresponding to the d children of P_j in the tree and let $m^1, \ldots, m^d$ be the messages they have to send in the basic protocol given the input vector $\vec{x}$. As claimed, for every $1 \leq k \leq d$, the message that the kth child sends in the modified protocol can be expressed as $s^k = m^k + (\vec{v}^k \cdot \vec{r})$. With this notation, to have $s^1 = w_1, \ldots, s^d = w_d$ the following linear system has to be satisfied:

$$\begin{bmatrix} \vec{v}^1 \\ \vdots \\ \vec{v}^d \end{bmatrix} \cdot \vec{r} = \begin{bmatrix} w_1 - m^1 \\ \vdots \\ w_d - m^d \end{bmatrix} .$$

Since $\vec{v}^1, \ldots, \vec{v}^d$ are linearly independent, this system has exactly one solution, as needed.

As for the root player the same argument can be applied to any fixed d-size subset of the $d+1$ messages it receives. This gives us that given any input vector $\vec{x}$, for all d-size messages vectors $\vec{w}$,

$$Pr[(s^1 = w_1) \wedge \cdots \wedge (s^d = w_d)] \;=\; \frac{1}{2^d} .$$

Now take two input vectors $\vec{x}$ and $\vec{y}$ such that $x_{\text{root}} = y_{\text{root}}$ and such that $f(\vec{x}) = f(\vec{y})$. Then by the correctness of the protocol, given a specific d-size messages vector, the $d+1$st message is the same for $\vec{x}$ and $\vec{y}$. Thus the privacy property holds with respect to the root also.

Finally, note that we assumed that the random choices were made by some external processor. However, we can let one of the leaf players randomly choose the bits $b_1, \ldots, b_d$ and supply each of the leaf players with the appropriate bit r_v. As the leaf players only send messages in the protocol, the special processor that selects the random bits gets no advantage.

Note that if a player is nonhonest it can easily prevent the other players from computing the correct output. However, it cannot get any additional information in the above protocol, since the only message each player gets after sending its own message is the value of the function. We have thus proved the following theorem.

THEOREM 3.1. *The function* `xor` *on n input bits can be computed privately using $d \geq 2$ random bits in $O(\log n / \log d)$ rounds.*

4. Lower bounds. In this section we prove several lower bounds on the number of rounds required to privately compute a boolean function, given that the total number of random bits the players can toss is d. The lower bound is given in terms of the sensitivity of the function. In section 4.1 we give some formal definitions. In section 4.2 we introduce the notion of *sensitivity* and present a lemma, central to our proofs, about sensitivity of functions. The proof of the lower bound appears in section 4.3.

4.1. Preliminaries. We first give a formal definition for the protocols. A protocol operates in rounds. In each round each player P_i, based on the value of its input bit x_i, the values of the messages it received in previous rounds, and the values of

the coins it tossed in previous rounds, tosses a certain number of additional coins and sends messages to the other players. The values of these messages may depend on all of the above, including the coins just tossed by P_i. The player may also choose to output the value of the function as calculated by itself (this is done only once by each player). Then, each player P_i receives the messages sent to it by the other players. To define the protocol more formally, we give the following definition.

DEFINITION 4.1 (view).

- *A time-t* partial *view of player P_i consists of its input bit x_i, the messages it received in the first $t-1$ rounds, and the coins it tossed in the first $t-1$ rounds. We denote it by $PView_i^t$.*
- *A time-t view of player P_i consists of its input bit x_i, the messages it received in the first $t-1$ rounds, and the coins it tossed in the first t rounds. We denote it by $View_i^t$.*

Intuitively, the *partial view* of a player P_i in round t determines how many coins (if at all) P_i will toss in round t. Then, its *view* in round t (which includes those newly tossed coins) determines, for round t, the messages that P_i will send and which value it will output (if at all) as the value of f. The formal definition of a protocol is given below.

DEFINITION 4.2. *A protocol consists of a set of functions $R_i^k : PView_i^k \to \mathcal{N}$ which determine how many coins are tossed by P_i in round k, and a set of functions $M_{i\to j}^k : View_i^k \to M$, $1 \le i, j \le n$ (where M is a finite domain of possible message values), which determine the message sent by P_i to P_j at round k.*

To quantify the amount of randomness used by a protocol we give the following definition.

DEFINITION 4.3. *A d-random protocol is a protocol such that for any input assignment, the total number of coins tossed by all players in* any *execution is at most d.*

Note that the definition allows that in different executions different players will toss the coins. This may depend on both the input of the players and previous coin tosses. Next we define the correctness of a protocol. We usually consider protocols that are always correct; protocols that are allowed to err will be considered in section 5.1.

DEFINITION 4.4. *A protocol to compute a function f is a protocol such that for any input vector $\vec{x}$ and every i, player P_i always correctly outputs the value of $f(\vec{x})$.*

It is sometime convenient to assume that each player P_i is provided with a random tape R_i, from which it reads random bits (rather than to assume that the player tosses random coins). The number of random coins tossed by player P_i is thus the rightmost position of this tape that it reads. We thus denote by R_i a specific random tape provided to player P_i, and denote by $\vec{R} = (R_1, \ldots, R_n)$ the vector or the random tapes of all the players ($\vec{r} = (r_1, \ldots, r_n)$ will denote the random variable for these tapes and vector of tapes). Note that if we fix $\vec{R}$, we obtain a deterministic protocol. Furthermore, $View_i^t$, for any i and t, is a function of the input assignment $\vec{x}$ and the random tapes of the players. We can thus write it as $View_i^t(\vec{x}, \vec{r})$. We denote by $T_i(\vec{x}, \vec{R})$ the round number in which player P_i outputs its result given input assignment $\vec{x}$ and random tapes for the players $\vec{R}$.

DEFINITION 4.5 (rounds complexity). *An r-round protocol to compute a function f is a protocol to compute f such that for all i, $\vec{x}$, $\vec{R}$, we have $T_i(\vec{x}, \vec{R}) \le r$.*

For the purpose of our proofs, we slightly modify our view of the protocol in the following way. Fix an arbitrary binary encoding for the messages in M. We will

consider a protocol where each player sends, instead of a single message from M, a set of boolean messages that represent the binary encoding of the message to be sent in the original protocol. These messages are sent "in parallel" in the same round. Hence, when we refer to messages we refer to these binary messages. Clearly, the number of rounds remains the same.

4.2. Sensitivity of functions. In this section we include some definitions related to functions $f : \{0,1\}^n \to D$, where D is some finite domain. Then, we present some useful properties related to these definitions.

DEFINITION 4.6 (sensitivity).

- *For a binary vector Y, denote by $Y^{(i)}$ the binary vector obtained from Y by flipping the ith entry.*
- *A function f is* sensitive *to its ith variable on assignment Y, if $f(Y) \neq f(Y^{(i)})$.*
- *$\mathcal{S}_f(Y)$ is the set of variables to which the function f is sensitive on assignment Y.*
- *The* sensitivity *of a function f, denoted $S(f)$, is $S(f) \stackrel{\triangle}{=} \max_Y |\mathcal{S}_f(Y)|$.*
- *The* average sensitivity *of a function f, denoted $AS(f)$, is the average of $|\mathcal{S}_f(Y)|$. That is, $AS(f) \stackrel{\triangle}{=} \frac{1}{2^n} \sum_{Y \in \{0,1\}^n} |\mathcal{S}_f(Y)|$.*
- *The set of variables on which f* depends, *denoted $\mathcal{D}(f)$, is $\mathcal{D}(f) \stackrel{\triangle}{=} \{i : \exists Y \text{ s.t. } i \in \mathcal{S}_f(Y)\}$. If $i \in \mathcal{D}(f)$ we say that f depends on its ith variable.*

The following claim gives a lower bound on the degree of error if we evaluate a function f by means of another function g, in terms of the average sensitivities of these functions. We use this property in our proofs.

CLAIM 4.7. *Consider any two functions $f, g : \{0,1\}^n \to D$. Then $f(\vec{x}) = g(\vec{x})$ for at most $2^n \cdot (1 - \frac{AS(f)-AS(g)}{2n})$ input assignments $\vec{x}$.*

Proof. Consider the n-dimensional hypercube. An *f-good edge* is an edge $e = (\vec{x}, \vec{y})$ such that $f(\vec{x}) \neq f(\vec{y})$. By the definitions, the number of f-good edges is exactly $\frac{2^n AS(f)}{2}$. Therefore, there are at least $2^n \frac{AS(f)-AS(g)}{2}$ edges which are f-good but not g-good. For each such edge $e = (\vec{x}, \vec{y})$ either $f(\vec{x}) \neq g(\vec{x})$ or $f(\vec{y}) \neq g(\vec{y})$. Since the degree of each vertex in the hypercube is n there must be at least $2^n \cdot \frac{AS(f)-AS(g)}{2n}$ inputs on which f and g do not agree. □

Next, we prove a lemma that bounds the growth of the sensitivity of a combination of functions. This lemma plays a central role in the proofs of our lower bounds, and any improvement on it will immediately improve our lower bounds.

LEMMA 4.8. *Let $\mathcal{F} = \{f_j\}, 1 \leq j \leq m$ be a set of m functions $f_j : \{0,1\}^n \to \{0,1\}$, for some n. Assume $S(f_j) \leq C$ for all j. Define the function $F(Y) \stackrel{\triangle}{=} (f_1(Y), \ldots, f_m(Y))$. If F assumes at most 2^d different values (different vectors), then the sensitivity of F is at most $C \cdot (2^d - 1)$.*[5]

Proof. Let Y be the assignment on which F has the largest sensitivity, i.e., $|\mathcal{S}_f(Y)| \geq |\mathcal{S}_f(Y')|$ for any assignment Y'. Without loss of generality, assume that $F(Y) = (0, \ldots, 0)$. Consider the set of neighbors of Y on which F has a value different than $(0, \ldots, 0)$ (the cardinality of this set is the sensitivity of F). There are at most $2^d - 1$ values of F attained on the assignments in this set. Consider one such value $q \in \{0,1\}^m$. There is at least one index j such that $q_j = 1$, and since the sensitivity of f_j is at most C, there can be at most C assignments $Y^{(i)}$ with the value q. We

[5] An obvious bound is $S(F) \leq C \cdot m$. However, for reasons that will become clear soon we are interested in bounds which are independent of m.

get that the total number of assignments $Y^{(i)}$ for which F has a value other than $(0,\ldots,0)$ is at most $C\cdot(2^d-1)$. □

4.3. Lower bound on the number of rounds. In this section we prove the following theorem.

THEOREM 4.9. *Let $\mathcal{A}$ be an r-round d-random ($d\geq 2$) private protocol to compute a boolean function f. Then, $r=\Omega(\log S(f)/d)$.*

The lower bound for the case $d=1$ is given in section 4.3.1. The first step of our proof uses the d-randomness property of the protocol to show that the number of views a player can see on a fixed input $\vec{x}$ is at most 2^d (over the different random tapes of all the players). Note that this is not obvious; although only d coins are tossed during every execution, the identity of the players that toss these coins may depend on the outcome of previous coin tosses.

LEMMA 4.10. *Consider a private d-random protocol to compute a boolean function f. Fix an input $\vec{x}$. Let $C_i^k(\vec{x},\vec{r})$ be the communication string seen by player P_i up to round k on input $\vec{x}$ and vector of random tapes $\vec{r}$. Then, for every player P_i, $C_i^k(\vec{x},\vec{r})$ can assume at most 2^d different values (over the different vectors of random tapes $\vec{r}$).*

Proof. For each execution we can order the coin tosses (i.e., readings from the local random tapes) according to the rounds of the protocol and within each round according to the index of the players that toss them. The identity of the player to toss the first coin is fixed by $\vec{x}$. The identity of the player to toss any next coin is determined by $\vec{x}$ and the outcome of the previous coins. Therefore, the different executions on input $\vec{x}$ can be described using the following binary tree: in each node of the tree we have a name of a player P_j that tosses a coin. The two outgoing edges from this node, labeled 0 and 1 according to the outcome of the coin, lead to two nodes labeled P_k and P_ℓ, respectively (k,ℓ, and j need not be distinct) which is the identity of the player to toss the next coin. If no additional coin toss occurs, the node is labeled "nil"; there are no outgoing edges from a nil node. By the d-randomness property of the protocol, the depth of the above tree is at most d; hence it has at most 2^d root-to-leaf paths. Every possible run of the protocol is described by one root-to-leaf path. Such a path determines all the messages sent in the protocol, which player tosses coins and when, and the outcome of these coins. In particular each path determines for any P_i the value of $C_i^k(\vec{x},\vec{r})$ (for any k). Hence, $C_i^k(\vec{x},\vec{r})$ can assume at most 2^d different values. □

In the following proof we restrict our attention to a specific deterministic protocol derived from the original protocol by fixing a specific vector of random tapes $\vec{R}=(R_1,\ldots,R_n)$ for the n players. In such a deterministic protocol the views of the players are functions of only the input assignment $\vec{x}$.

LEMMA 4.11. *Consider a private d-random protocol to compute a boolean function f. Fix random tapes $\vec{R}=(R_1,\ldots,R_n)$. Recall that $View_i^k(\vec{x},\vec{r})$ is the view of player P_i at round k on input $\vec{x}$ and vector of random tapes $\vec{r}$. Then, for any P_i, $View_i^k(\vec{y},\vec{R})$ can assume at most 2^{d+2} different values (over the values of $\vec{y}$).*

Proof. Partition the input assignments $\vec{x}$ into 4 groups according to the value of x_i (0 or 1) and the value of $f(\vec{x})$ (0 or 1). We argue that the number of different values the view can assume within each such group is at most 2^d. Fix an input $\vec{x}$ in one of these four groups and consider any other input $\vec{y}$ pertaining to the same group. Recall that $C_i^k(\vec{y},\vec{R})$ is the communication string seen by player P_i until round k on input $\vec{y}$ and when the random tapes of the players are $\vec{R}$. If the value of $C_i^k(\vec{y},\vec{R})$ is some

communication string C_i, then by the privacy requirement[6], communication C_i must also occur by round k when the input is $\vec{x}$, and the vector of random tapes is some $\vec{R}' = (R'_1, \ldots, R'_n)$, where $R'_i = R_i$. Thus, the value of $C_i^k(\vec{y}, \vec{R})$ must also appear as $C_i^k(\vec{x}, \vec{r})$ for some vector of random tapes. However, by Lemma 4.10, $C_i^k(\vec{x}, \vec{r})$ can assume at most 2^d values (over the values of $\vec{r}$). Thus, $C_i^k(\vec{y}, \vec{R})$ can assume at most 2^d values over the possible input assignments that pertain to the same group.

Now, observe that $View_i^k(\vec{y}, \vec{r})$ is determined by the input bit y_i, the communication string $C_i^k(\vec{y}, \vec{r})$, and the random tape r_i. Therefore, on $\vec{R}$ and on two input assignments $\vec{y}$ and $\vec{y}'$ of the same group (in particular $y_i = y'_i$), if $View_i^k(\vec{y}, \vec{R}) \neq View_i^k(\vec{y}', \vec{R})$, then $C_i^k(\vec{y}, \vec{R}) \neq C_i^k(\vec{y}', \vec{R})$. Thus, $View_i^k(\vec{y}, \vec{R})$ can assume at most 2^d different values over the input assignments that pertain to the same group. □

The following lemma gives an upper bound on the sensitivity of the view of a player at a given round, in terms of the number of random bits and the round number. This will enable us to give a lower bound on the number of necessary rounds.

LEMMA 4.12. *Consider a private d-random protocol to compute a boolean function f, and consider a specific vector of random tapes $\vec{R}$, and the deterministic protocol derived by it. Then for every player P_i, the function $View_i^k(\vec{x}, \vec{R})$ (as a function of $\vec{x}$ only) has sensitivity of at most $Q(k) \stackrel{\triangle}{=} (2^{d+2})^{k-1}$.*

Proof. First note that since we fix the random tapes, the views of the players are functions of the input assignment $\vec{x}$ only. We prove the lemma by induction. For $k = 1$ the view of any player depends only on its single input bit. Thus, the claim is obvious. For $k > 1$ assume the claim holds for any $\ell < k$. This implies, in particular, that all messages received by player P_i and included in the view under consideration have sensitivity of at most $Q(k-1)$. Clearly, the input bit itself has sensitivity 1 which is at most $Q(k-1)$. Thus, the view under consideration is composed of bits each having sensitivity at most $Q(k-1)$. Moreover, by Lemma 4.11 the view can assume at most 2^{d+2} values. It follows from Lemma 4.8 that the sensitivity of the view under consideration is at most $Q(k-1) \cdot (2^{d+2} - 1) \leq Q(k)$. (Note that Lemma 4.8 allows us to give a bound which does not depend on the *number* of messages received by P_i.) □

We can now give the lower bound on the number of rounds, in terms of the sensitivity of the function and the number of random bits.

THEOREM 4.13. *Given a private d-random protocol ($d \geq 2$) to compute a boolean function f, consider the deterministic protocol derived from it by any given random tapes $\vec{R}$. For any player P_i, there is at least one input assignment $\vec{x}$ such that $T_i(\vec{x}, \vec{R}) = \Omega(\log S(f)/d)$.*

Proof. Consider a fixed but arbitrary player P_i. Denote by t the largest round number in which P_i outputs a value, i.e., $t = \max_{\vec{x}}\{T_i(\vec{x}, \vec{R})\}$. We claim that as long as the sensitivity of the view of P_i does not reach $S(f)$, there is at least one input assignment for which P_i cannot output the correct value of f. Let Y be an input assignment on which the sensitivity $S(f)$ is obtained. That is, the value of $F(Y)$ is different than the value of F on $S(f)$ of Y's "neighbors." Hence, if the sensitivity of the view of P_i is less than $S(f)$, then the output of P_i must be wrong on either Y or on at least one of these "neighbors" (as the sensitivity of the view is an upper bound on the sensitivity of the output). Thus, t is such that $S(View_i^t(\vec{x}, \vec{R})) \geq S(f)$. By Lemma 4.12, we get $2^{(d+2)(t-1)} \geq S(f)$, i.e., $t \geq \frac{\log S(f)}{(d+2)} + 1$. □

[6] The privacy requirement is defined on the final communication string, but this clearly implies the same requirement on any prefix of it.

This proves Theorem 4.9; moreover, it shows not only that there is an input assignment $\vec{x}$ and random tapes $\vec{R}$ for which the protocol runs "for a long time," but also that for each vector of random tapes $\vec{R}$ there is such input assignment. The following corollary follows for the function `xor` (using the fact that $S(\texttt{xor}) = n$).

COROLLARY 4.14. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits. Then $r = \Omega(\log n/d)$.*

4.3.1. Lower bound for a single random bit ($d = 1$). For the case of a single random bit ($d = 1$), we have the following lower bound.

THEOREM 4.15. *Let $\mathcal{A}$ be an r-round 1-random private protocol to compute a boolean function f. Then, $r = \Omega(S(f))$.*

To prove the theorem, we restrict our attention to one of the two deterministic protocols derived from the original protocol by fixing the value of the random bit[7]. The messages and views in this protocol are functions of the input vector, $\vec{x}$, only. Let Y be an assignment on which $S(f)$, the sensitivity of f, is obtained. For a given function m, a variables x_j is called *good* for m on Y if both m and f are sensitive to x_j on Y. We denote by $\mathcal{G}_m(Y)$ the set of good variables on Y, i.e., $\mathcal{G}_m(Y) \stackrel{\triangle}{=} \mathcal{S}_m(Y) \cap \mathcal{S}_f(Y)$. We first prove the following two lemmas.

LEMMA 4.16. *Consider any player P_i. Denote by m_1 a message that P_i receives such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$. Then for any other message m_2 received by P_i such that $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, either* (a) $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} = \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$, *or* (b) $|\mathcal{G}_{m_1}(Y) \cup \mathcal{G}_{m_2}(Y)| \geq S(f) - 1$.

Proof. Assume, toward a contradiction, that both (a) and (b) do not hold. First, since $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$ and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$ there are two variables $x_k \in \mathcal{G}_{m_1}(Y)$, $x_\ell \in \mathcal{G}_{m_2}(Y)$ such that $k \neq i$ and $\ell \neq i$. Moreover, by the assumption that (a) does not hold, we can assume, without loss of generality, (as to the names of m_1 and m_2) that $x_\ell \notin \mathcal{G}_{m_1}(Y)$ (in particular, $\ell \neq k$). By the assumption that (b) does not hold, there is a variable x_j ($j \neq i$) such that f is sensitive to x_j on Y, but both m_1 and m_2 are not sensitive to x_j on Y. Now consider the following three input assignments:

- $Y_0 = Y^{(j)}$.
- $Y_1 = Y^{(k)}$.
- $Y_2 = Y^{(\ell)}$.

Consider $View_i$ on the above three inputs and assume, without loss of generality, that $m_1(Y) = m_2(Y) = 0$. Since both m_1 and m_2 are *not* sensitive to x_j on Y, then $m_1(Y^{(j)}) = 0$ and $m_2(Y^{(j)}) = 0$. Since m_1 is sensitive to x_k on Y, then $m_1(Y^{(k)}) = 1$. Since m_2 is sensitive to x_ℓ on Y, but m_1 is not, then $m_1(Y^{(\ell)}) = 0$ and $m_2(Y^{(\ell)}) = 1$. Hence, $View_i$ assumes three different values for Y_0, Y_1, and Y_2. The function f is sensitive on Y to all of j, k, and ℓ; therefore, $f(Y_0) = f(Y_1) = f(Y_2)$, and x_i is equal in all three assignments. However, in the proof of Lemma 4.11, it is shown that the number of values of $View_i$ corresponding to inputs with the same value of f and the same value of x_i is at most $2^d = 2$—a contradiction. □

The following lemma gives an upper bound on the sensitivity of the view of the player in terms of the round number. We then use this lemma to give a lower bound on the number of necessary rounds.

LEMMA 4.17. *Let $t \leq (S(f) - 1)/2$ be a round number and P_i be any player. Then, $|\mathcal{G}_{View_i^t}(Y)| \leq t$.*

[7] We let the identity of the player that tosses this coin possibly depend on the input $\vec{x}$. However, note that if we want the privacy and 1-randomness properties to hold, this cannot be the case.

Proof. We prove the claim by induction on t. For $t = 1$, clearly $|\mathcal{G}_{View_i^t}(Y)| \leq 1$ (since before getting any messages, the view depends only on x_i). For $1 < t \leq (S(f)-1)/2$ assume the claim holds for any $k < t$. Denote by M the set of messages received by P_i and included in the view under consideration. Clearly $\mathcal{G}_{View_i^t}(Y) \subseteq \{x_i\} \cup (\cup_{m \in M} \mathcal{G}_m(Y))$. There could be one of three cases:

1. For any message $m \in M$, $|\mathcal{G}_m(Y) \setminus \{x_i\}| = 0$. In this case the claim clearly holds.
2. Any two messages $m_1, m_2 \in M$, such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$, and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, satisfy $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} = \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$. It follows that $|\mathcal{G}_{View_i^t}(Y)| \leq |\{x_i\} \cup (\cup_{m \in M} \mathcal{G}_m(Y))| \leq |\mathcal{G}_{m_1}(Y)| + 1$. Since by the induction hypothesis $|\mathcal{G}_{m_1}(Y)| < t$, then $|\mathcal{G}_{View_i^t}(Y)| \leq t$.
3. There are two messages $m_1, m_2 \in M$, such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$, and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, but $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} \neq \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$. By Lemma 4.16, $|\mathcal{G}_{m_1}(Y) \cup \mathcal{G}_{m_2}(Y)| \geq S(f) - 1$, and (without loss of generality) $|\mathcal{G}_{m_1}(Y)| \geq (S(f)-1)/2$. This contradicts the induction hypothesis as m_1 was received in some round $k < t \leq (S(f)-1)/2$ and therefore generated by a view of round k. By the induction hypothesis, $|\mathcal{G}_{m_1}(Y)| \leq k < (S(f)-1)/2$. □

We can now give the proof of Theorem 4.15.

Proof of Theorem 4.15. Consider any player P_i. Denote by t the largest round number in which P_i outputs a value, i.e., $t = \max_{\vec{x}}\{T_i(\vec{x}, 0)\}$. As in the proof of Theorem 4.13, it must be that $|\mathcal{G}_{View_i^t}(Y)| \geq S(f)$, and therefore, by Lemma 4.17 we have $t > (S(f)-1)/2$. □

For the function `xor` we have the following corollary.

COROLLARY 4.18. *Let $\mathcal{A}$ be an r-round 1-random private protocol to compute* `xor` *of n bits. Then $r = \Omega(n)$.*

5. Lower bounds on the expected number of rounds. As the protocols we consider are randomized, it is possible that for the same input $\vec{x}$, different random tapes for the players will result in executions that run for different number of rounds. Hence, it is natural to consider not only the *worst-case* running time but also the *expected* running time. Usually, saying that a protocol has expected running time r means that for every input $\vec{x}$ the expected time until *all* players finish the execution is bounded by r (where the expectation is computed over the choices of the random tapes of the players). Here we consider a weaker definition, which requires only the *existence* of a player i whose expected running time is bounded by r. As we are proving a lower bound, this only makes our result stronger: it would mean that for *every* player there is an input assignment for which the expected running time is high. Note that it is not necessarily the case that the first player that computes the value of the function can announce this value (and thus all players compute the value within one round). The reason is that the fact that a certain player computes the function at a certain round may reveal some information on the inputs, and hence such announcing may violate the privacy requirement (see [CGK90]). We first define the expected rounds complexity of a protocol.

DEFINITION 5.1 (expected rounds complexity). *An* expected *r-round protocol to compute a function f is a protocol to compute f such that there exists a player P_i such that for all $\vec{x}$, $E_{\vec{r}}[T_i(\vec{x}, \vec{r})] \leq r$.*

The lower bound that we prove in this section is in terms of the average sensitivity of the computed function. In particular, we prove an $\Omega(\log n/d)$ lower bound on the expected number of rounds required by protocols that privately compute `xor` of n bits. We will prove the following theorem.

THEOREM 5.2. *Let f be a boolean function and let $\mathcal{A}$ be an* expected *r-round d-random private protocol ($d \geq 2$) to compute the function f. Then, $r = \Omega(AS(f) \cdot \log AS(f)/nd)$.*

To prove the theorem we consider a protocol $\mathcal{A}$ and fix any player P_i. We say that the protocol is *late* on input $\vec{x}$ and vector of random tapes $\vec{R}$ if $T_i(\vec{x}, \vec{R}) \geq \frac{\log AS(f)}{2(d+2)} + 1$. We define a $0-1$ random variable $L(\vec{x}, \vec{r})$ to be 1 if and only if the protocol is late on $\vec{x}$ and $\vec{r}$. For the purpose of our proofs in this section we define a uniform distribution on the 2^n input assignments (this is not to say that the input is actually drawn by such distribution). Moreover, note that the domain of vectors of random tapes is enumerable.

We first show that for any deterministic protocol derived from a private protocol to compute f, not only is there at least one input on which the protocol is late, but this happens for a large fraction of the inputs.

LEMMA 5.3. *Consider a player P_i and any fixed vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$. Then*

$$E_{\vec{x}}[L(\vec{x}, \vec{R})] \geq \frac{AS(f) - \sqrt{AS(f)}}{2n} \ .$$

Proof. Consider the views of P_i, $View_i^t$, given the vector of random tapes $\vec{R}$. For any round t such that $t < \frac{\log AS(f)}{2(d+2)} + 1$, by Lemma 4.12, we get $S(View_i^t) < 2^{(d+2)\frac{\log AS(f)}{2(d+2)}} = \sqrt{AS(f)}$. Any function g computed from such a view can have at most the same sensitivity, and thus clearly an average sensitivity of at most $\sqrt{AS(f)}$. By Claim 4.7, such a function g can have the correct value for the function f for at most $2^n(1 - \frac{AS(f)-\sqrt{AS(f)}}{2n})$ input assignments. Since we assume that $\mathcal{A}$ is correct for *all* input assignments, it follows that at least $2^n \frac{AS(f)-\sqrt{AS(f)}}{2n}$ nput assignments are late. □

We can now give a lower bound on the expected number of rounds.

LEMMA 5.4. *Consider a player P_i. There is at least one input assignment $\vec{x}$ for which*

$$E_{\vec{r}}[T_i(\vec{x}, \vec{r})] \geq \left(\frac{AS(f) - \sqrt{AS(f)}}{2n}\right)\left(\frac{\log AS(f)}{2(d+2)} + 1\right) = \Omega(AS(f) \cdot \log AS(f)/nd) \ .$$

Proof. By Lemma 5.3, $E_{\vec{r},\vec{x}}[L(\vec{x}, \vec{r})] \geq \frac{AS(f)-\sqrt{AS(f)}}{2n}$. Hence, there is at least one input assignment $\vec{x}$ for which $E_{\vec{r}}[L(\vec{x}, \vec{r})] \geq \frac{AS(f)-\sqrt{AS(f)}}{2n}$. For such $\vec{x}$ we get

$$E_{\vec{r}}[T_i(\vec{x}, \vec{r})] \geq \left(\frac{AS(f) - \sqrt{AS(f)}}{2n}\right) \cdot \left(\frac{\log AS(f)}{2(d+2)} + 1\right),$$

as needed. □

Theorem 5.2 follows from the above lemma. The following corollary applies to the function `xor`.

COROLLARY 5.5. *Let $\mathcal{A}$ be an* expected *r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits. Then, $r = \Omega(\log n/d)$.*

Proof. The proof follows from Theorem 5.2 and the fact that $AS(\texttt{xor}) = n$. □

5.1. Weakly correct protocols. In this section we consider protocols that are allowed to make a certain amount of errors. Given a protocol $\mathcal{A}$, denote by $\mathcal{A}_i(\vec{x},\vec{r})$ the output of the protocol in player P_i, given input assignment $\vec{x}$ and vector of random tapes $\vec{r}=(r_1,\ldots,r_n)$.

DEFINITION 5.6. *For $\delta<1/2$, a $(1-\delta)$-correct protocol to compute a function f is a protocol that, for every player P_i and every input vector $\vec{x}$, satisfies $Pr_{\vec{r}}[\mathcal{A}_i(\vec{x},\vec{r})=f(\vec{x})]\geq(1-\delta)$.*

Note that while designing a protocol one usually wants a stronger requirement; that is, with high probability *all* players compute the correct value. With the above definition, it is possible that in every execution of the protocol at least one of the players is wrong. However, as our aim now is to prove a lower bound this weak definition only makes our result stronger.

In the following theorem we give lower bounds on the number of rounds and on the expected number of rounds for weakly correct protocols.

THEOREM 5.7. *Let f be a boolean function.*

- *Let $\mathcal{A}$ be a $(1-\delta)$-correct r-round, d-random private protocol ($d\geq 2$) to compute f. If $\delta<\frac{AS(f)-\sqrt{AS(f)}}{2n}$ then $r=\Omega(\log AS(f)/d)$.*
- *Let $\mathcal{A}$ be a $(1-\delta)$-correct* expected *r-round, d-random private protocol ($d\geq 2$) to compute f. Then $r=\Omega((1-\sqrt{2\delta})\cdot(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\delta/2})\cdot\frac{\log AS(f)}{d})$.*

Proof. We first prove the lower bound on the number of rounds and then turn our attention to the expected number of rounds. The correctness requirement implies that for any player P_i, $Pr_{\vec{r}}[\mathcal{A}_i(\vec{x},\vec{r})=f(\vec{x})]\geq 1-\delta$, for all $\vec{x}$. This implies that there exists a vector of random tapes $\vec{R}$ such that for at least $2^n(1-\delta)$ input assignments $\vec{x}$, $\mathcal{A}_i(\vec{x},\vec{R})=f(\vec{x})$. As in the proof of Lemma 5.3 (using Claim 4.7), it follows that before round number $\frac{\log AS(f)}{2(d+2)}+1$, the protocol can be correct on at most $2^n(1-\frac{AS(f)-\sqrt{AS(f)}}{2n})$ inputs (with random tapes $\vec{R}$). Since we require that at least $2^n(1-\delta)$ are correct, we have that at least

$$2^n(1-\delta)-2^n\left(1-\frac{AS(f)-\sqrt{AS(f)}}{2n}\right)=2^n\left(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\delta\right)$$

inputs are late. To get a lower bound on r for an r-round protocol, it is sufficient to have a single input vector $\vec{x}$ such that the execution on $(\vec{x},\vec{R})$ is "long." For this, note that if $\delta<\frac{AS(f)-\sqrt{AS(f)}}{2n}$, then (for random tapes $\vec{R}$) the number of late inputs is greater than 0. This gives us a lower bound of $r=\Omega(\log AS(f)/d)$ for any $(1-\delta)$-correct r-round, d-random protocol, with δ as above.

We now turn to the lower bound on the *expected* number of rounds of $(1-\delta)$-correct protocols. Consider a player P_i. Define a $0-1$ random variable $G(\vec{x},\vec{r})$ to be 1 if and only if $\mathcal{A}_i(\vec{x},\vec{r})=f(\vec{x})$. Then, the correctness requirement implies that $E_{\vec{r}}[G(\vec{x},\vec{r})]\geq 1-\delta$, for all $\vec{x}$. It follows that for any $\vec{R}$ the probability that $\vec{R}$ satisfies $E_{\vec{x}}[G(\vec{x},\vec{R})]\geq 1-\sqrt{\delta/2}$ is at least $1-\sqrt{2\delta}$.[8] For any such vector of random tapes $\vec{R}$, consider the deterministic protocol derived from it. In such a protocol there are

[8] Otherwise, $E_{\vec{r},\vec{x}}[G(\vec{x},\vec{r})]<(1-\sqrt{2\delta})\cdot 1+\sqrt{2\delta}\cdot(1-\sqrt{\delta/2})=1-\sqrt{2\delta}+\sqrt{2\delta}-\delta=1-\delta$. Thus there is at least one input assignment $\vec{x}$ such that $E_{\vec{r}}[G(\vec{x},\vec{r})]<1-\delta$, which is a contradiction to the protocol being $1-\delta$-correct.

at least

$$2^n\left(1-\sqrt{\frac{\delta}{2}}\right)-2^n\left(1-\frac{AS(f)-\sqrt{AS(f)}}{2n}\right)=2^n\left(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\frac{\delta}{2}}\right)$$

late input assignments; that is, $E_{\vec{x}}[L(\vec{x},\vec{R})] \geq (\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\delta/2})$. Thus,

$$E_{\vec{r},\vec{x}}[L(\vec{x},\vec{r})] \geq (1-\sqrt{2\delta})\cdot\left(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\frac{\delta}{2}}\right) .$$

It follows that there is at least one input assignment $\vec{x}$ for which

$$E_{\vec{r}}[L(\vec{x},\vec{r})] \geq (1-\sqrt{2\delta})\cdot\left(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\frac{\delta}{2}}\right) ,$$

which implies that

$$E_{\vec{r}}[T_i(\vec{x},\vec{r})] \geq (1-\sqrt{2\delta})\cdot\left(\frac{AS(f)-\sqrt{AS(f)}}{2n}-\sqrt{\frac{\delta}{2}}\right)\cdot\frac{\log AS(f)}{d},$$

as claimed. □

The following gives the lower bounds for the function `xor`.

COROLLARY 5.8. *For fixed $\delta < 1/2$ let $\mathcal{A}$ be a $(1-\delta)$-correct d-random* expected *r-round private protocol to compute* `xor` *of n bits. Then $r = \Omega(\log n/d)$. (Obviously the same lower bound holds for r-round protocols.)*

Proof. The proof follows from Theorem 5.7 and the fact that $AS(\texttt{xor}) = n$. Note that the expression $(1-\sqrt{2\delta})(\frac{1}{2}-\sqrt{\delta/2}-\frac{1}{2\sqrt{n}})$ is greater than 0 for any $\delta < 1/2$ (and sufficiently large n). □

6. Conclusion. In this paper we initiate the quantitative study of randomness in private computations. As mentioned in the introduction, our work was already followed by additional work on this topic [BDPV95, KM96, KOR96, CKOR97].

We give upper and lower bounds on the number of rounds required for computing `xor` privately with a given number of random bits. Alternatively, we give bounds on the number of random bits required for computing `xor` privately within a given number of rounds. Our lower bounds extend to other functions in terms of their sensitivity (and average sensitivity).

An obvious open problem is to close the gap between the upper bound and the lower bound for computing `xor` using d-random bits. One possible way of doing this is to improve the bound given by Lemma 4.8.

Appendix. Improved lower bounds for some special cases.

A.1. A special case. Each player sends a single message. In section 3 we show that `xor` can be computed privately with d random bits, in $O(\log n/\log d)$ rounds. Corollary 4.14 shows a lower bound of $\Omega(\log n/d)$ rounds for such a computation. In this section we prove a stronger lower bound than the one proved in section 4, but in a weaker model. In this model, each player is allowed to send a single nonconstant message. More precisely, each player sends only a single nonconstant message to a specific other player, and this other player is the same in all runs. Note that the protocol presented in section 3 has this property.

Again, we consider the protocol obtained by fixing a *specific* vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$. The main observation is that the above property implies that if a player receives two messages, then the sets of variables on which the two messages depend are disjoint.

In the following we prove an extension of Lemma 4.12, which gives an upper bound on the sensitivity of the view of the players. To do so, we first prove a stronger version of Lemma 4.8 for the case that each function depends on a different set of variables. We then give a slight variation of the proof of Lemma 4.12, using the new lemma instead of Lemma 4.8.

LEMMA A.1. *Let $\mathcal{F} = \{f_j\}, 1 \leq j \leq m$ be a set of m functions $f_j : \{0,1\}^n \to \{0,1\}$, for some n. Assume $S(f_j) \leq C$ for all j. Further assume that, for all $i \neq j$, $\mathcal{D}(f_i) \cap \mathcal{D}(f_j) = \emptyset$. Define $F(Y) = (f_1(Y), \ldots, f_m(Y))$. If F assumes at most 2^d different values (different vectors), then the sensitivity of F is at most $C \cdot d$.*

Proof. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of functions which are not constant functions. Let $m' = |\mathcal{F}|$ be the cardinality of this set. Since for every $i \neq j$ such that $f_i, f_j \in \mathcal{F}'$, we have $\mathcal{D}(f_i) \cap \mathcal{D}(f_j) = \emptyset$, then the number of values assumed by F is $2^{m'}$. Therefore $m' \leq d$.

Let Y be the assignment on which F has the largest sensitivity, i.e., $|\mathcal{S}_f(Y)| \geq |\mathcal{S}_f(Y')|$ for any assignment Y'. If F is sensitive to i on Y, then there is some function $f_j \in \mathcal{F}'$ such that f_j is sensitive to i on Y. However every function f_j has sensitivity at most C. Therefore F is sensitive on Y to at most $C \cdot m' \leq C \cdot d$ variables. □

We remark that the bound proved by the above lemma is tight.

LEMMA A.2. *Consider a private d-random protocol to compute a boolean function f with the additional property under consideration, and consider a specific vector of random tapes $\vec{R}$. Then $View_i^k(\vec{x}, \vec{R})$, for any player P_i, has sensitivity of at most $Q(k) \stackrel{\triangle}{=} (d+1)^{k-1}$.*

Proof. First note that, since we fix the random tapes, the views of the players are functions of the input assignment $\vec{x}$ only. We prove the lemma by induction. For $k = 1$ the view of any player depends only on its single input bit. Thus the claim is obvious. For $k > 1$ assume the claim holds for any $\ell < k$. This implies in particular that all messages received by player P_i and included in the view under consideration have sensitivity of at most $Q(k-1)$. Consider the view without the input bit x_i and denote it by F; this is composed of only the messages received by P_i and the bits read from the (local) random tape. Using Lemma 4.11, and Lemma A.1 with $C = Q(k-1)$, we have that F has sensitivity at most $Q(k-1) \cdot d = (d+1)^{k-2} \cdot d$. The view under consideration, $View_i^k(\vec{x}, \vec{R})$, has an additional single bit, which clearly has sensitivity at most 1. We thus have that the sensitivity of $View_i^k(\vec{x}, \vec{R})$ is at most $(d+1)^{k-2} \cdot d + 1 \leq Q(k)$. □

THEOREM A.3. *Given a private d-random protocol ($d \geq 2$) to compute a boolean function f that has the special property under consideration, consider the deterministic protocol derived from it by any given vector of random tapes $\vec{R}$. For every player P_i, there is at least one input assignment $\vec{x}$ such that $T_i(\vec{x}, \vec{R}) = \Omega(\log S(f)/\log d)$.*

The proof follows the proof of Theorem 4.13, using Lemma A.2 instead of Lemma 4.12. The lower bound of this section is stated in the following corollary.

COROLLARY A.4. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute a boolean function f that has the property under consideration. Then $r = \Omega(\log S(f)/\log d)$.*

The following lower bound applies for the function `xor`.

COROLLARY A.5. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$)*

to compute `xor` *of* n *bits that has the property under consideration. Then* $r = \Omega(\log n/\log d)$.

A.2. A special case. Protocols with messages of a special type. As proved in section 3, `xor` can be computed privately with d-random bits in $O(\log n/\log d)$ rounds. For general protocols, Corollary 4.14 shows a lower bound of $\Omega(\log n/d)$ rounds, while in a special case (in which the upper bound falls) Corollary A.5 gives a tight $\Omega(\log n/\log d)$ lower bound on the number of rounds. Here we consider another special case in which each message m can be expressed as a boolean function $f(\vec{x}) \oplus g(\vec{r})$, of the n-entry input vector $\vec{x}$, and a d-entry binary random tape, $\vec{r}$. Again, our upper bound satisfies this restriction, and the lower bound we prove in this case is tight (i.e., $\Omega(\log n/\log d)$). One can think of such protocols as those which are designed (as we do in section 3) by first designing a nonprivate protocol, and then "masking" the messages with random noise (which is canceled at the end). In particular, each player can toss all of its coins before the protocol starts (the number of coins tossed may depend, however, on the outcome of previous coin tosses). The particular masking bit that is used for each message is not to depend on the input to the players.

The idea of the proof is to consider the number of input variables which are *good* (with respect to the computed function f) in the view of a player at some round k (as in section 4.3.1).[9] Clearly, the input variable of the player may be good. Other good variables can be those that are good for any of the messages that this player received in a previous round. The key ingredient in the proof will be to show that the number of messages received by any player, that are "beneficial" to increase the number of good variables in its view, is at most d (otherwise, the privacy requirement is violated). From this we get that the number of good variables at round k is $O((d+1)^{k-1})$. In order for a player to output the correct value of f its view must have at least $S(f)$ good variables. The result will follow.

More precisely, consider a private protocol (with the property under consideration) to compute a boolean function f. Consider the messages $M_1, \ldots, M_m$ received by some player P_j during the protocol. By assumption, we can write each of these messages, M_ℓ, as $M_\ell(\vec{x}, \vec{r}) = f_\ell(\vec{x}) \oplus g_\ell(\vec{r})$. Denote by Y an input assignment on which f achieves its sensitivity $S(f)$. Let $\mathcal{G}_{M_\ell}(Y)$ denote the set of *good* variables of the message M_ℓ on Y; henceforth we denote *good* to mean "good on Y." Note that for any message M_ℓ, due to the special structure of messages in this protocol, $\mathcal{G}_{M_\ell}(Y) = \mathcal{G}_{f_\ell}(Y)$. First, ignore those messages M_ℓ for which $\mathcal{G}_{M_\ell}(Y) \subseteq \{x_j\}$. Assume that the rest of the messages are ordered in a way such that each message M_i (where $i \leq m' \leq m$) has at least one good variable that is not good for any of $M_1, \ldots, M_{i-1}$ (i.e., $\mathcal{G}_{M_i}(Y) \setminus \cup_{i'<i}\mathcal{G}_{M_{i'}}(Y) \neq \emptyset$). Further assume that $|\cup_{i=1}^{m'} \mathcal{G}_{M_i}(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$. We first present a claim which gives a property of $g_1, \ldots, g_{m'}$. This claim is then used to prove that $m' \leq d$.

CLAIM A.6. *Let* $M_1, \ldots, M_{m'}$ *be as above, and let* $g_1, \ldots, g_{m'}$ *be the corresponding* g_i*'s. Then, the vector* $(g_1(\vec{r}), \ldots, g_{m'}(\vec{r}))$ *assumes all* $2^{m'}$ *values (over the choices of the random tapes* $\vec{r}$*).*

Proof. Suppose this is not the case. Then there exists a minimal k such that $(g_1(\vec{r}), \ldots, g_{k-1}(\vec{r}))$ assumes all 2^{k-1} values, but $(g_1(\vec{r}), \ldots, g_k(\vec{r}))$ does not assume all 2^k values. In particular, there are $k-1$ bits $a_1, \ldots, a_{k-1}$ for which (without loss of

[9] Let Y be an assignment and f be the computed function. Recall that for a given function m, a variable x_j is called *good* for m on Y if both m and f are sensitive to x_j on Y.

generality) the values vector $(a_1, \ldots, a_{k-1}, 0)$ is assumed but $(a_1, \ldots, a_{k-1}, 1)$ is not.

By the assumptions on the M_i's, there exists a variable x_{i_k}, $i_k \neq j$, such that x_{i_k} is good for M_k (and hence for f_k), but is not good for any of $f_1, \ldots, f_{k-1}$. This implies that both f and f_k are sensitive to x_{i_k} (on Y). In particular, $f_k(Y) \neq f_k(Y^{(x_{i_k})})$. On the other hand, for any $\ell < k$, $f_\ell(Y) = f_\ell(Y^{(x_{i_k})})$. Since $|\cup_{i=1}^{m'} \mathcal{G}_{M_i}(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$, there is a variable x_s, $s \neq j$, such that f is sensitive to x_s on Y, but f_i, $i \leq k$ are not.

Now consider the two input assignments $Y^{(x_s)}$ and $Y^{(x_{i_k})}$. Without loss of generality, assume that $(f_1(Y), \ldots, f_k(Y)) = (0, \ldots, 0)$. Then, $(f_1(Y^{(x_s)}), \ldots, f_k(Y^{(x_s)})) = (0, \ldots, 0)$, and $(f_1(Y^{(x_{i_k})}), \ldots, f_k(Y^{(x_{i_k})})) = (0, \ldots, 0, 1)$. Therefore, on the assignment $Y^{(x_s)}$, player P_j can see (with a positive probability) the vector of messages $(a_1, \ldots, a_{k-1}, 0)$, but not $(a_1, \ldots, a_{k-1}, 1)$, and vice versa for the input assignment $Y^{(x_{i_k})}$. Since $f(Y^{(x_{i_k})}) = f(Y^{(x_s)})$, and x_j is equal in both of them, the privacy requirement (with respect to P_j) is violated. □

The important consequence of the above claim is that $m' \leq d$. This is because, by Lemma 4.10, there are at most 2^d different vectors for $(g_1(\vec{r}), \ldots, g_{m'}(\vec{r}))$. We are now ready to prove the theorem.

THEOREM A.7. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute a boolean function f, such that $\mathcal{A}$ has the property that each message M_ℓ can be expressed as $f_\ell(\vec{x}) \oplus g_\ell(\vec{r})$. Then $r = \Omega(\log S(f)/\log d)$.*

Proof. Recall that Y is an input assignment on which f has sensitivity $S(f)$ and that by "good" we mean "good on Y." We first prove by induction that the view of any player in round k, on input Y, can have at most $(d+1)^{k-1}$ good variables, or at least $S(f)$ good variables. This is certainly true in the first round, where the view can have at most one good variable. Now consider the view of P_j in round k. Denote by M the set of messages received by the player. If the view has less than $S(f)$ good variables, then $|\cup_{m \in M} \mathcal{G}_m(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$. By Claim A.6, it follows that there can be at most d messages, each of which has at least one good variable which is not x_j, and such that each has at least one such good variable that the previous ones (in the order we defined) do not have. Each of these messages was received in one of the previous rounds, and has less than $S(f)$ good variables; hence, by the induction hypothesis each has at most $(d+1)^{k-2}$ good variables. Thus, the view of P_j at round k can have at most $d \cdot (d+1)^{k-2} + 1 \leq (d+1)^{k-1}$ good variables.

To prove the theorem, fix some vector of random tapes $\vec{R}$ and consider the views obtained when running the deterministic protocol derived by it. As remarked before, due to the special type of message, the set of good variables is the same for all random tapes. Obviously, the view of a player that outputs the value f must have $S(f)$ good variables (on Y). Let t be the *first* round in which some player P_j has view with at least $S(f)$ good variables. Order the messages received by P_j in round t in an arbitrary order. Then, there is the first message m after which the view of P_j has at least $S(f)$ good variables. This message m has at most $(d+1)^{t-2}$ good variables, by the above claim (as m was produced from a view of a previous round, which has less than $S(f)$ good variables). On the other hand, just before this message is received, the view of the player does not have $S(f)$ good variables; therefore, by the above claim, it has at most $(d+1)^{t-1}$ good variables. We thus have

$$S(f) \leq (d+1)^{t-1} + (d+1)^{t-2} ,$$

which gives the required lower bound on t. □

COROLLARY A.8. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$)*

to compute `xor` *of* n *bits that has the property under consideration. Then* $r = \Omega(\log n / \log d)$.

Acknowledgments. We thank Gábor Tardos for improving the constant and simplifying the proof of Lemma 4.8 and Demetrios Achlioptas for his help in simplifying the proof of Lemma A.1. We also thank Benny Chor for useful comments.

REFERENCES

[AGHP90] N. Alon, O. Goldreich, J. Håstad, and R. Peralta, *Simple constructions of almost k-wise independent random variables*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 544–553; also in Random Structures Algorithms, 3 (1992), pp. 289–304; also see addendum in 4 (1993), pp. 119–120.

[BB89] J. Bar-Ilan and D. Beaver, *Non-cryptographic fault-tolerant computing in a constant number of rounds*, in Proc. 8th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1989, pp. 201–209.

[B89] D. Beaver, *Perfect Privacy for Two-Party Protocols*, Technical Report TR-11-89, Harvard University, 1989.

[BFKR90] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, *Security with low communication overhead*, in Advances in Cryptology, Proc. Crypto '90, Lecture Notes in Computer Science 537, Springer-Verlag, New York, 1991, pp. 62–76.

[BGG90] M. Bellare, O. Goldreich, and S. Goldwasser, *Randomness in interactive proofs*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 563–571.

[BGW88] M. Ben-or, S. Goldwasser, and A. Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, in Proc. 20th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1988, pp. 1–10.

[BM84] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, in Proc. 22nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1982, pp. 112–117; also in SIAM J. Comput., 13 (1984), pp. 850–864.

[BGS94] C. Blundo, A. Giorgio Gaggia, and D. R. Stinson, *On the dealer's randomness required in secret sharing schemes*, in Proc. EuroCrypt94, Lecture Notes in Computer Science 950, Springer-Verlag, New York, 1995, pp. 35–46.

[BSV94] C. Blundo, A. De-Santis, and U. Vaccaro, *Randomness in distribution protocols*, in Proc. 21st Annual Intl. Collegium on Automata, Languages, and Programming, Lecture Notes in Computer Science 820, Springer-Verlag, New York, 1994, pp. 568–579.

[BDPV95] C. Blundo, A. De-Santis, G. Persiano, and U. Vaccaro, *On the number of random bits in totally private computations*, in Proc. 22nd Annual Intl. Collegium on Automata, Languages, and Programming, Lecture Notes in Computer Science 944, Springer-Verlag, New York, 1995, pp. 171–182.

[CG90] R. Canetti and O. Goldreich, *Bounds on tradeoffs between randomness and communication complexity*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 35–44; also in Comput. Complexity, 3 (1993), pp. 141–167.

[CKOR97] R. Canetti, E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Randomness vs. fault-tolerance*, in Proc. 16th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1997, pp. .

[CCD88] D. Chaum, C. Crepeau, and I. Damgard, *Multiparty unconditionally secure protocols*, in Proc. 20th Annual ACM Symposium on Theory of Computing, ACM, New York, 1988, pp. 11–19.

[CK89] B. Chor and E. Kushilevitz, *A zero-one law for boolean privacy*, in Proc. 21st Annual ACM Symposium on Theory of Computing, ACM, New York, 1989, pp. 62–72; also in SIAM J. Discrete Math., 4 (1991), pp. 36–47.

[CK92] B. Chor and E. Kushilevitz, *A communication-privacy tradeoff for modular addition*, Inform. Process. Lett., 45 (1993), pp. 205–210.

[CGK90] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *Private computations over the integers*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer

Science, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 335–344; also in SIAM J. Comput., 24 (1995), pp. 376–386.

[CGK92] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *On the structure of the privacy hierarchy*, J. Cryptology, 7 (1994), pp. 53–60.

[CG88] B. Chor and O. Goldreich, *Unbiased bits from sources of weak randomness and probabilistic communication complexity*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 429–442; also in SIAM J. Comput., 17 (1988), 230–261.

[CW89] A. Cohen and A. Wigderson, *Dispersers, deterministic amplification, and weak random sources*, in Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 14–19.

[FY92] M. Franklin and M. Yung, *Communication complexity of secure computation*, in Proc. 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 699–710.

[IZ89] R. Impagliazzo and D. Zuckerman, *How to recycle random bits*, in Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 248–253.

[KK94] D. Karger and D. Koller, *(De)randomized construction of small sample spaces in NC*, in Proc. 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 252–263.

[KM93] D. Koller and N. Megiddo, *Constructing small sample spaces satisfying given constraints*, in Proc. 25th Annual ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 268–277.

[KM94] H. Karloff and Y. Mansour, *On construction of k-wise independent random variables*, in Proc. 26th Annual ACM Symposium on Theory of Computing, ACM, New York, 1993, pp. 564–573.

[KPU88] D. Krizanc, D. Peleg, and E. Upfal, *A time-randomness tradeoff for oblivious routing*, in Proc. 20th Annual ACM Symposium on Theory of Computing, ACM, New York, 1988, pp. 93–102.

[K89] E. Kushilevitz, *Privacy and communication complexity*, in Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1989, pp. 416–421; also in SIAM J. Discrete Math., 5 (1992), pp. 273–284.

[KM96] E. Kushilevitz and Y. Mansour, *Randomness in private computations*, in Proc. 15th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1996, pp. 181–190; also in SIAM J. Discrete Math., 10 (1997), pp. 647–661.

[KMO94] E. Kushilevitz, S. Micali, and R. Ostrovsky, *Reducibility and completeness in multi-party private computations*, in Proc. 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 478–489.

[KOR96] E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Characterizing linear size circuits in terms of privacy*, in Proc. 28th Annual ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 541–550.

[NN90] J. Naor and M. Naor, *Small-bias probability spaces: Efficient constructions and applications*, in Proc. 22nd Annual ACM Symposium on Theory of Computing, ACM, New York; also in SIAM J. Comput., 22 (1993), pp. 838–856.

[N90] N. Nisan, *Pseudorandom generator for space bounded computation*, in Proc. 22nd Annual ACM Symposium on Theory of Computing, ACM, New York, 1990, pp. 204–212.

[RS89] P. Raghavan and M. Snir, *Memory vs. randomization in on-line algorithms*, in Proc. 16th Annual Intl. Collegium on Automata, Languages, and Programming, Lecture Notes in Computer Science 372, Springer-Verlag, Berlin, 1989, pp. 687–703.

[S92] L. J. Schulman, *Sample spaces uniform on neighborhoods*, in Proc. 24th Annual ACM Syposium on Theory of Computing, ACM, New York, 1992, pp. 17–25.

[VV85] U. Vazirani and V. Vazirani, *Random polynomial time is equal to slightly-random polynomial time*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 417–428.

[Y82] A. C. Yao, *Theory and applications of trapdoor functions*, in Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1982, pp. 80–91.

[Z91] D. Zuckerman, *Simulating BPP using a general weak random source*, in Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 79–89.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 81–93, February 1998

006

SOMETIMES TRAVELLING IS EASY: THE MASTER TOUR PROBLEM*

VLADIMIR G. DEĭNEKO†, RÜDIGER RUDOLF†, AND GERHARD J. WOEGINGER†

Abstract. In 1975, Kalmanson proved that if the distance matrix in the travelling salesman problem (TSP) fulfills certain combinatorial conditions (that are nowadays called the *Kalmanson conditions*) then the TSP is solvable in polynomial time [*Canad. J. Math.*, 27 (1995), pp. 1000–1010].

We deal with the problem of deciding, for a given instance of the TSP, whether there is a renumbering of the cities such that the corresponding renumbered distance matrix fulfills the Kalmanson conditions. Two results are derived: first, it is shown that—in case it exists—such a renumbering can be found in polynomial time. Secondly, it is proved that such a renumbering exists if and only if the instance possesses the so-called *master tour* property. A recently posed question by Papadimitriou is thereby answered in the negative.

Key words. travelling salesman problem, Kalmanson condition, master tour, combinatorial optimization

AMS subject classifications. 08C85, 68Q20, 05C38, 68R10, 90C35

PII. S0895480195281878

1. Introduction.

The *travelling salesman problem* (TSP) is defined as follows. Given an $n \times n$ distance matrix $C = (c_{ij})$, find a permutation $\pi \in S_n$ that minimizes the sum $\sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)}$. In other words, the salesman must visit cities 1 to n in arbitrary order and want to minimize the total travel length. This problem is one of the fundamental problems in combinatorial optimization and known to be NP hard. For more specific information on the TSP, the reader is referred to the book by Lawler et al. [7].

In this paper, we are interested in a special case of the TSP where—due to special combinatorial structures in the distance matrix—the problem is solvable in polynomial time: the case of *Kalmanson* distance matrices. A symmetric $n \times n$ matrix C is called a *Kalmanson* matrix if it fulfills the conditions

$$c_{ij} + c_{k\ell} \leq c_{ik} + c_{j\ell} \text{ for all } 1 \leq i < j < k < \ell \leq n, \tag{1.1}$$

$$c_{i\ell} + c_{jk} \leq c_{ik} + c_{j\ell} \text{ for all } 1 \leq i < j < k < \ell \leq n. \tag{1.2}$$

Note that these conditions do not involve any diagonal entries c_{ii}. Since every city is visited only once, diagonal entries are of no relevance for the TSP and may as well be considered to be "undefined" or zero. Originally, Kalmanson introduced these conditions in order to generalize the concept of *convexity* of finite point sets in the plane: for some convex planar point set, let $p_1, \ldots, p_n$ denote its clockwise ordering around the convex hull. Then the Euclidean distance matrix $c_{ij} = d(p_i, p_j)$ fulfills all conditions (1.1) and (1.2) (Proof: In a convex quadrangle, the total length of the diagonals is greater or equal to the total length of two opposite sides.) Moreover, if we "rotate" the ordering by one point, the distance matrix of the resulting *rotated* point

*Received by the editors February 21, 1995; accepted for publication (in revised form) January 6, 1996. This research has been supported by the Spezialforschungsbereich F 003 "Optimierung und Kontrolle," Projektbereich Diskrete Optimierung.

http://www.siam.org/journals/sidma/11-1/28187.html

†Institut für Mathematik B, Steyrergasse 30, A-8010 Graz, Austria (deineko@opt.math.tu-graz.ac.at, rudolf@opt.math.tu-graz.ac.at, gwoegi@opt.math.tu-graz.ac.at).

sequence $p_2, p_3, \ldots, p_n, p_1$ also is a Kalmanson matrix. It is easy to verify that this "rotation property" does not result from special Euclidean features but solely from inequalities (1.1) and (1.2). Hence, if one removes the first row and first column from a Kalmanson matrix and appends them after the last row and column, the result of this operation is another Kalmanson matrix. Similarly, reversing the ordering of the rows and columns of a Kalmanson matrix will again yield a Kalmanson matrix.

Kalmanson [6] proved that for the TSP with a Kalmanson distance matrix, the identity permutation $\langle 1, 2, 3, \ldots, n\rangle$ always constitutes an optimal tour and thus, the TSP is easily solved for this special case. Observe that the length of the optimum TSP tour is not changed when the cities are renumbered, i.e., when the rows and columns of the distance matrix are permuted according to the same permutation. However, such a renumbering will usually destroy the Kalmanson conditions. Intuition tells us that a renumbered instance is still a rather trivial special case of the TSP, since it is just a Kalmanson instance in disguise, but it is by no means obvious how to recognize this disguise. Hence, the problem arises of finding a permutation that transforms the distance matrix back into a Kalmanson matrix.

Another problem related to the TSP is the detection of a master tour, motivated by the following observation. Suppose that all cities in a Euclidean instance of the TSP are the vertices of a convex polygon. Then the optimum tour is not only easy to find (it is the perimeter of the polygon), but the instance also fulfills the much stronger *master tour* property: there is an optimum TSP tour π such that the optimum TSP tour of any subset of cities can be obtained by simply omitting from the tour π the cities that are not in the subset. Such a tour π is called a master tour. The concept of a master tour was first formulated by Papadimitriou [8, 9]. It is easy to prove that deciding whether a given instance of the TSP has the master tour property is in the complexity class $\Sigma_2\mathbf{P}$. Papadimitriou also considered the corresponding decision problem as a "good candidate for a natural $\Sigma_2\mathbf{P}$-complete problem." In this paper, we will prove that the following results hold true.

(1) For a symmetric $n \times n$ matrix C, it can be decided in $O(n^2 \log n)$ time whether C is a permuted Kalmanson matrix.
(2) A distance matrix allows a master tour if and only if it is a permuted Kalmanson matrix.

Combining results (1) and (2) yields a polynomial-time algorithm for the master tour problem. Hence, unless $\Sigma_2\mathbf{P}=\mathbf{P}$, the conjecture of Papadimitriou is false.

Organization of the paper. Section 2 summarizes elementary definitions and results on permutations and matrices. In section 3, several lemmas on the combinatorial structure of Kalmanson matrices are collected. These lemmas are used in section 4 to derive an $O(n^2 \log n)$-time algorithm for recognizing permuted $n \times n$ Kalmanson matrices. Section 5 explains the connection between permuted Kalmanson matrices and master tours and shows that a master tour can be detected in polynomial time. Finally, section 6 closes with a short discussion.

2. Definitions and preliminaries. In this section, several basic definitions for permutations and matrices are summarized.

For an $n \times n$ matrix C, denote by $I = \{1, \ldots, n\}$ the set of rows (columns). A row i *precedes* a row j in C ($i \prec j$ for short), if row i occurs before row j in C. For two sets K_1 and K_2 of rows, we write $K_1 \prec K_2$ if and only if $k_1 \prec k_2$ for all $k_1 \in K_1$ and $k_2 \in K_2$. Let $V = \{v_1, v_2, \ldots, v_r\}$ and $W = \{w_1, w_2, \ldots, w_s\}$ be two subsets of I. We denote by $C[V, W]$ the $r \times s$ submatrix of C that is obtained by deleting all rows not contained in V and all columns not in W.

For permutations, we adopt the notation $\pi = \langle x_1, x_2, \ldots, x_n \rangle$ for "$\pi(i) = x_i$ for $1 \le i \le n$." The *concatenation* of permutations $\langle x_1, \ldots, x_n \rangle$ and $\langle y_1, \ldots, y_m \rangle$ is $\langle z_1, \ldots, z_{n+m} \rangle$, where $z_i = x_i$ for $1 \le i \le n$ and $z_{n+j} = y_j$ for $1 \le j \le m$. The identity permutation is denoted by ε, i.e., $\varepsilon(i) = i$ for all $i \in I$. For a permutation ϕ, the permutation ϕ^- defined by $\phi^-(i) = \phi(n - i + 1)$ is called the *reverse permutation* of ϕ. Permutation ϕ is called a *cyclic shift* or a *rotation* if there exists a $k \in I$ such that $\phi = \langle k, k+1, \ldots, n, 1, \ldots, k-1 \rangle$.

By $C_{\phi,\pi}$ we denote the matrix which is obtained from matrix C by permuting its rows according to ϕ and its columns according to π, i.e., $C_{\phi,\pi} = (c_{\phi(i),\pi(j)})$. For $C_{\phi,\phi}$, we usually write C_ϕ. A permutation ϕ is called a *Kalmanson permutation for some matrix* C if C_ϕ is a Kalmanson matrix. A matrix C is called a *permuted Kalmanson matrix* if there exists a Kalmanson permutation for C.

For a partition $V = \langle V_1, \ldots, V_v \rangle$ of I into v subsets, the set $\textsc{Str}(V_1, \ldots, V_v)$ is defined to contain all permutations ϕ that fulfill $\phi(v_i) \prec \phi(v_j)$ for all $v_i \in V_i$ and $v_j \in V_j$ with $1 \le i < j \le v$. $\textsc{Str}(V_1, \ldots, V_v)$ is called the set of permutations induced by the sequence of *stripes* $V_1, \ldots, V_v$. An appropriate data structure for storing, manipulating, and intersecting such sets of permutations are *PQ trees* as introduced by Booth and Lueker [1] (in fact, PQ trees of *height two* suffice to represent these permutations).

PROPOSITION 2.1 (see Booth and Lueker [1]). *For two partitions $\langle U_1, \ldots, U_u \rangle$ and $\langle V_1, \ldots, V_v \rangle$ of I, the set $\textsc{Str}(U_1, \ldots, U_u) \cap \textsc{Str}(V_1, \ldots, V_v)$ either equals $\textsc{Str}(W_1, \ldots, W_w)$ for an appropriate partition $W = \langle W_1, \ldots, W_w \rangle$ of I or it is empty. The partition W can be computed in $O(|I|)$ time.*

An $m \times n$ matrix C is called a *sum-matrix* if there exist numbers $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$ such that $c_{ij} = x_i + y_j$ for all i and j. Note that this implies $c_{ij} + c_{rs} = c_{is} + c_{rj}$ for $1 \le i < r \le m$ and $1 \le j < s \le n$ (i.e. in any two by two submatrix, both diagonals have equal sums). For convenience, single rows and columns are also considered to be sum-matrices. Note that every sum-matrix is a Kalmanson matrix and a Contra Monge matrix. An $m \times n$ matrix C is called a *Contra Monge matrix* if $c_{ij} + c_{rs} \ge c_{is} + c_{rj}$ holds for $1 \le i < r \le m$ and $1 \le j < s \le n$. The combinatorial structure of Contra Monge matrices and of permuted Contra Monge matrices is well understood (see the original paper by Deĭneko and Filonenko [4] or the survey paper by Burkard, Klinz, and Rudolf [2]). The known main results are summarized in the following proposition.

PROPOSITION 2.2. *Let $X = (x_{ij})$ be an $m \times n$ matrix. Let $\Pi \subseteq S_m \times S_n$ denote the set of all pairs of permutations (π, ϕ) such that $X_{\pi,\phi}$ is a Contra Monge matrix.*

(i) *Then either Π is the empty set, or there exists an appropriate partition $R_1, \ldots, R_r$ of the set R of rows and an appropriate partition $C_1, \ldots, C_c$ of the set C of columns of X such that*

$$\Pi = \{(\pi, \phi) | \pi \in \Pi_R, \phi \in \Pi_C\} \cup \{(\pi, \phi) | \pi^- \in \Pi_R, \phi^- \in \Pi_C\}$$

where $\Pi_R = \textsc{Str}(R_1, \ldots, R_r)$ and $\Pi_C = \textsc{Str}(C_1, \ldots, C_c)$.

(ii) *The partitions $R_1, \ldots, R_r$ and $C_1, \ldots, C_c$ can be computed in $O(mn + m \log m + n \log n)$ time (in case they exist).*

(iii) *Every submatrix $X[R_i, C]$ and every submatrix $X[R, C_j]$ is a sum-matrix. These sum–matrices are maximal sum–matrices in X (i.e., neither rows nor columns may be added without destroying the sum-matrix property).*

(iv) *In case Π is not empty, either $r = c = 1$ holds (and X is a sum-matrix), or the numbers r and c are both at least two (and X is horizontally and vertically divided into several stripes by Π).*

(v) *Matrix X is a Contra Monge matrix if and only if for all pairs of indices $1 \le i \le m-1$ and $1 \le j \le n-1$, the inequality*

$$x_{ij} + x_{i+1,j+1} \ge x_{i,j+1} + x_{i+1,j} \tag{2.1}$$

is fulfilled.

By $\mathbb{K}$ we denote the set of Kalmanson matrices. Similarly to the above alternate characterization (2.1) of Contra Monge matrices, an alternate characterization of Kalmanson matrices can be given.

PROPOSITION 2.3. *An $n \times n$ symmetric matrix C is a Kalmanson matrix if*

$$c_{i,j+1} + c_{i+1,j} \le c_{ij} + c_{i+1,j+1} \quad \text{for all } 1 \le i \le n-3,\ i+2 \le j \le n-1, \tag{2.2}$$

$$c_{i,1} + c_{i+1,n} \le c_{in} + c_{i+1,1} \quad \text{for all } 2 \le i \le n-2. \tag{2.3}$$

Observe that conditions (2.2) and (2.3) can be verified in $O(n^2)$ time. This yields the following proposition.

PROPOSITION 2.4. *For a symmetric $n \times n$ matrix C, it can be decided in $O(n^2)$ time whether C is a Kalmanson matrix.*

PROPOSITION 2.5. *Let C be an $n \times n$ Kalmanson matrix, let $I' \subseteq I$, and let $\phi \in S_n$ be a cyclic shift. Then* (i) $C_{\varepsilon^-} \in \mathbb{K}$, (ii) $C[I', I'] \in \mathbb{K}$, *and* (iii) $C_\phi \in \mathbb{K}$ *hold.*

3. Combinatorial properties of Kalmanson matrices. In this section, we derive several technical lemmas on the combinatorial structure of Kalmanson matrices.

LEMMA 3.1. *Let C be an $n \times n$ symmetric Kalmanson matrix, $2 \le m \le n-1$, let $V = \langle 1, 2, \ldots, m \rangle$, and let $W = \langle m+1, \ldots, n \rangle$. Then $C[V, W]$ is a Contra Monge matrix.*

Proof. This is a consequence of condition (2.2) in Proposition 2.3. □

LEMMA 3.2. *Let C be a symmetric $n \times n$ matrix. Let V and W be a partition of I with $|V| = r \ge 2$, $|W| = s \ge 2$ such that $C[V, W]$ is a sum-matrix. Let $q \in W$ and $p \in V$ be arbitrary. Let $D = C[V \cup \{q\}, V \cup \{q\}]$ and $E = C[\{p\} \cup W, \{p\} \cup W]$. Assume that there is a permutation $\psi = \langle v_1, \ldots, v_r, q \rangle$ of $V \cup \{q\}$ and a permutation $\pi = \langle p, w_1, \ldots, w_s \rangle$ of $W \cup \{p\}$ such that $D_\psi \in \mathbb{K}$ and $E_\pi \in \mathbb{K}$.*

Under these conditions either $C_\phi \in \mathbb{K}$ for $\phi = \langle v_1, \ldots, v_r, w_1, \ldots, w_s \rangle$ or there does not exist any permutation $\sigma \in \mathrm{STR}(V, W)$ with $C_\sigma \in \mathbb{K}$.

Proof. We prove that under the conditions in the lemma either the inequality

$$c_{v_1 v_r} + c_{w_1 w_s} \le c_{v_1 w_1} + c_{v_r w_s} \tag{3.1}$$

is fulfilled and $C_\phi \in \mathbb{K}$ for $\phi = \langle v_1, \ldots, v_r, w_1, \ldots, w_s \rangle$ or inequality (3.1) is not fulfilled and there does not exist any permutation $\sigma \in \mathrm{STR}(V, W)$ with $C_\sigma \in \mathbb{K}$.

First assume that inequality (3.1) is fulfilled. We prove that $C_\phi \in \mathbb{K}$ according to Proposition 2.3 by verifying conditions (2.2) and (2.3). Consider two indices i and j in C_ϕ, $1 \le i \le n-3$, and $i+2 \le j \le n-1$. In case $i \prec i+1 \prec j \prec j+1$ are all in V or are all in W, condition (2.2) holds for C_ϕ since $D_\psi \in \mathbb{K}$ and $E_\pi \in \mathbb{K}$. If i and $i+1$ are in V and j and $j+1$ are in W, the four elements $c_{i,j+1}$, $c_{i+1,j}$, c_{ij}, and $c_{i+1,j+1}$ lie in the sum-matrix $C[V, W]$ and thus trivially fulfill (2.2). Next, if i is in V and $i+1 \prec j \prec j+1$ are in W then $i = v_r$ and $i+1 = w_1$ holds. The relations $p \prec w_1 \prec j \prec j+1$ in $E_\pi \in \mathbb{K}$ yield $c_{p,j+1} + c_{w_1,j} \le c_{pj} + c_{w_1,j+1}$. Since $p, v_r \in V$, $j, j+1 \in W$, and $C[V, W]$ is a sum-matrix, $c_{pj} + c_{v_r,j+1} = c_{p,j+1} + c_{v_r,j}$. Adding this equality to the previous inequality yields (2.2). The last case where $i \prec i+1 \prec j$ are

in V and $j+1$ is in W is handled symmetrically. Summarizing, (2.2) is true in any case.

Next, consider an index $2 \le i \le n-2$. In case $i \neq v_r$, (2.3) is true since $D_\psi \in \mathbb{K}$ (respectively, $E_\pi \in \mathbb{K}$) holds. In case $i = v_r$, (2.3) is exactly (3.1). Hence, (3.1) implies (2.3), and the first half of the lemma is proven.

To prove the remaining half, assume that for some $\sigma \in \textsc{Str}(V,W)$, $C_\sigma \in \mathbb{K}$ holds. We show how to derive inequality (3.1) from this. Since V precedes W in σ, only two cases arise:

(i) $v_1 \prec v_r \prec w_1 \prec w_s$ or $v_r \prec v_1 \prec w_s \prec w_1$ in σ. Then condition (1.1) yields (3.1).

(ii) $v_1 \prec v_r \prec w_s \prec w_1$ or $v_r \prec v_1 \prec w_1 \prec w_s$ in σ. Then condition (1.1) yields $c_{v_1v_r} + c_{w_1w_s} \le c_{v_rw_1} + c_{v_1w_s}$. Since $C[V,W]$ is a sum-matrix, $c_{v_1w_s} + c_{v_rw_1} = c_{v_1w_1} + c_{v_rw_s}$. Adding these two inequalities gives (3.1). □

LEMMA 3.3. *Let C be a symmetric $n \times n$ matrix. Let $U_1, \dots, U_m$ be a partition of I such that $C[U_i, I \setminus U_i]$ is a sum-matrix for $1 \le i \le m$. Let u_i be an arbitrary element in U_i. Let π_i be a Kalmanson permutation for $C[U_i \cup \{u_{i+1}\}, U_i \cup \{u_{i+1}\}]$ (indices are taken modulo m, i.e., $u_{m+1} = u_1$) that has u_{i+1} as its last element. Let ϕ_i denote the permutation of U_i induced by π_i.*

Under these conditions either $C_\phi \in \mathbb{K}$ where ϕ is the concatenation of $\phi_1, \dots, \phi_m$ or there does not exist any Kalmanson permutation for C in $\textsc{Str}(U_1, \dots, U_m)$.

Proof. The proof is done by induction on the number t of stripes U_i with cardinality at least two. If $t = 0$, the statement trivially holds. Otherwise if $t \ge 1$, we may assume without loss of generality that $|U_1| \ge 2$. Moreover, we assume that there exists a Kalmanson permutation for C in $\textsc{Str}(U_1, \dots, U_m)$, since otherwise there is nothing to show.

Set $W = U_2 \cup \dots \cup U_m$ and consider the sets U_i' where $U_1' = \{u_1\}$ and $U_i' = U_i$ for $2 \le i \le m$. By the induction assumption, the concatenation of $\langle u_1 \rangle, \phi_2, \dots, \phi_m$ is a Kalmanson permutation for $C[\{u_1\} \cup W, \{u_1\} \cup W]$. Set $V = U_1$. By the conditions of the lemma, the concatenation π_1 of ϕ_1 and $\langle u_2 \rangle$ is a Kalmanson permutation for $C[V \cup \{u_2\}, V \cup \{u_2\}]$. Moreover, the matrix $C[V,W]$ is a sum-matrix. Summarizing, all conditions for applying Lemma 3.2 with $p = u_1$ and $q = u_2$ are fulfilled. The statement in Lemma 3.2 yields that the concatenation ϕ is indeed a Kalmanson permutation and the inductive proof is complete. □

The following notation is convenient. For two rows i and j of a matrix C, define the set

$$\mathcal{M}(i,j) = \big\{ k \in I \setminus \{i,j\} \mid c_{ik} - c_{jk} = \min_{\ell \neq i,j} \{c_{i\ell} - c_{j\ell}\} \big\}. \tag{3.2}$$

Note that $C[\{i,j\}, \mathcal{M}(i,j)]$ is a sum matrix. In case $|\mathcal{M}(i,j)| = n-2$ holds, $c_{ik} - c_{jk} =$ const for all $k \in I \setminus \{i,j\}$. Such a pair of rows is called *equivalent*, and this is denoted by $i \sim j$. We also define that every row is equivalent to itself.

LEMMA 3.4. *For any symmetric $n \times n$ matrix C, the relation $\sim$ is an equivalence relation.*

Proof. By definition, the relation $\sim$ is symmetric and reflexive. To prove that $\sim$ is transitive, consider $i, j_1, j_2 \in I$ with $j_1 \sim i$ and $i \sim j_2$. The goal is to show that $j_1 \sim j_2$, i.e., to show that for any $k, \ell \in I$ with $\{j_1, j_2\} \cap \{k, \ell\} = \emptyset$ the equality $(*)$ $c_{j_1k} - c_{j_2k} = c_{j_1\ell} - c_{j_2\ell}$ holds. If $i \notin \{k, \ell\}$ holds, we use $j_1 \sim i$ and $j_2 \sim i$: subtract the equalities $c_{ik} - c_{j_1k} = c_{i\ell} - c_{j_1\ell}$ and $c_{ik} - c_{j_2k} = c_{i\ell} - c_{j_2\ell}$ from each other, and derive $(*)$. Otherwise, assume that $k = i$. Use $i \sim j_2$ to obtain $c_{ij_1} - c_{j_2j_1} = c_{i\ell} - c_{j_2\ell}$

and use $i \sim j_1$ to obtain $c_{ij_2} - c_{j_1 j_2} = c_{i\ell} - c_{j_1 \ell}$. Subtracting these equations yields $(*)$. □

LEMMA 3.5. *Let C be a symmetric $n \times n$ matrix. If $1 \sim i$ for all $i \in I$, then $C \in \mathbb{K}$.*

Proof. By Lemma 3.4 above, all inequalities (1.1) and (1.2) are fulfilled with equality. □

LEMMA 3.6. *Let C be a symmetric $n \times n$ Kalmanson matrix. Let i and j be two rows of C with $i \prec j$, let $K_1 = \mathcal{M}(i,j) \cup \{i\}$, and let $K_2 = I \setminus K_1$. Then there exists a cyclic shift ϕ such that $C_\phi \in \mathbb{K}$ and $K_1 \prec K_2$ in C_ϕ.*

Proof. By definition, $i \in K_1$ and $j \in K_2$. Consider any $k \in \mathcal{M}(i,j)$. Then $c_{ik} - c_{jk} = c_{i\ell} - c_{j\ell}$ for all $\ell \in K_1 \setminus \{i\}$ and $c_{ik} - c_{jk} < c_{i\ell} - c_{j\ell}$ for all $\ell \in K_2 \setminus \{j\}$. We distinguish the following three cases on the relative positions of i, j, and k in C.

(i) $k \prec i \prec j$. Then condition (1.2) yields $c_{ip} - c_{jp} \leq c_{ik} - c_{jk}$ for any p with $k \prec p \prec i \prec j$. Hence, $p \in K_1$ for all $p \in I$ with $k \prec p \prec i$.

(ii) $i \prec k \prec j$. Analogously to the argument in (i), condition (1.1) implies $p \in K_1$ for all $p \in I$ with $i \prec p \prec k$.

(iii) $i \prec j \prec k$. Analogously to the argument in (i), conditions (1.2) and (1.2) imply $p \in K_1$ for all $p \in I$ with $p \prec i$ or $k \prec p$.

Summarizing, we conclude that there exist two elements r and s such that either $K_1 = \{r, \ldots, i, \ldots, s\}$ or $K_2 = \{s+1, \ldots, j, \ldots, r-1\}$. By Proposition 2.5(iii), every cyclic shift of C again yields a Kalmanson matrix. Hence, choosing $\phi = \langle r, \ldots, s, \ldots, n, 1, \ldots, r-1 \rangle$ or choosing $\phi = \langle r, \ldots, n, 1, \ldots, s, s+1, \ldots, r-1 \rangle$ completes the argument. □

4. Recognition of permuted Kalmanson matrices. This section shows how to recognize permuted Kalmanson matrices in polynomial time. The recognition algorithm is described in two steps: first we give a rough outline of the algorithm in subsection 4.1. We sketch a divide and conquer approach that is based on the lemmas derived in the preceding section. Then in subsection 4.2, we describe a fast implementation of the algorithm that runs in $O(n^2 \log n)$ time.

4.1. Outline of the algorithm. Given an $n \times n$ matrix C, we want to decide whether there exists a permutation σ such that $C_\sigma \in \mathbb{K}$ and we want to compute σ in case it exists. Our solution algorithm follows a divide and conquer strategy. The main goal is to find in polynomial time $D(n)$ a so-called *nice bipartition* of the set I of rows, i.e., a bipartition into two sets V and W that satisfies the following three properties.

(N1) $|V|, |W| \geq 2$.

(N2) $C[V,W]$ is a sum-matrix.

(N3) The matrix C is a permuted Kalmanson matrix if and only if there exists a permutation $\sigma \in \text{STR}(V,W)$ with $C_\sigma \in \mathbb{K}$.

If we have found some nice bipartition, we choose rows $q \in W$ and $p \in V$ and recursively compute Kalmanson permutations ψ and π for the two matrices $C[V \cup \{q\}, V \cup \{q\}]$ and $C[\{p\} \cup W, \{p\} \cup W]$. According to Lemma 3.2 and property (N3) above, either the concatenation of ψ and π is a Kalmanson permutation for C, or C cannot be a permuted Kalmanson matrix. By Proposition 2.4, it can be decided in $O(n^2)$ time whether the concatenation of ψ and π indeed yields a Kalmanson permutation. Summarizing, this results in a recursive algorithm with time complexity

$$T(n) \leq \max_{2 \leq k \leq n-2} \{T(k+1) + T(n-k+1)\} + D(n) + O(n^2). \tag{4.1}$$

1. Find a row k with $k \not\sim 1$.
   If k does not exist: $\Longrightarrow$ C itself is Kalmanson matrix. Stop.
2. From k, define an initial partition of I with two stripes K_1 and K_2.
3. If all stripes in the current partition of I have cardinality one:
   $\Longrightarrow$ Only one potential Kalmanson permutation left. Stop.
4. Rotate the current partition such that the first stripe has cardinality at least two.
5. If the first stripe in the current partition of I together with its complement forms a nice bipartition: $\Longrightarrow$ Nice bipartition found. Stop.
6. Refine the partition by applying Proposition 2.2 to the submatrix whose row set is the first stripe and whose column set is the complement of the first stripe: $\Longrightarrow$ Refinement of partition found. Goto 3.

FIG. 4.1. *A high-level description of how to find a nice bipartition.*

It is easy to verify that $T(n) = O(nD(n) + n^3)$ and hence, the algorithm runs in polynomial time. It remains to explain how to find a nice bipartition (a high-level pseudocode description of this procedure is given in Figure 4.1).

First, we find a row k that is *not* equivalent to row 1 (if such a row k does not exist, if follows from Lemma 3.5 that the identity permutation is a Kalmanson permutation). Compute $\mathcal{M}(1,k)$ and define the sets $K_1 = \mathcal{M}(1,k) \cup \{1\}$ and $K_2 = I \setminus K_1$. Since $1 \not\sim k$, $|K_1|, |K_2| \geq 2$ holds. By Lemma 3.6, it is sufficient to deal with permutations ϕ for which $K_1 \prec K_2$ holds, i.e., with permutations $\phi \in \textsc{Str}(K_1, K_2)$. Now, if $C[K_1, K_2]$ is a sum-matrix, K_1 and K_2 form a nice bipartition and we are done. Otherwise, by Lemma 3.1, it is necessary to deal with permutations ϕ for which the matrix $C_\phi[K_1, K_2]$ is a Contra Monge matrix. According to Proposition 2.2, these permutations can be described by $\phi \in \textsc{Str}_1 \cup \textsc{Str}_1^*$ where $\textsc{Str}_1 = \textsc{Str}(K_{11}, \ldots, K_{1r}, K_{21}, \ldots, K_{2s})$ and $\textsc{Str}_1^* = \textsc{Str}(K_{1r}, \ldots, K_{11}, K_{2s}, \ldots, K_{21})$ for appropriate partitions $K_{11}, \ldots, K_{1r}$ of K_1 and $K_{21}, \ldots, K_{2s}$ of K_2. It is easy to see that by rotation and reversion, every $\phi^* \in \textsc{Str}_1^*$ can be transformed into some $\phi \in \textsc{Str}_1$. By Proposition 2.5 we conclude that in case C can be permuted into a Kalmanson matrix; this can also be reached by some $\phi \in \textsc{Str}_1$ and thus it is sufficient to consider permutations in $\textsc{Str}_1$.

In case all stripes in $\textsc{Str}_1$ have cardinality one, there remains just a single potential Kalmanson permutation (and it can be checked in $O(n^2)$ time whether the permutation indeed is a Kalmanson permutation). Otherwise, there is some stripe K_{ij} of cardinality at least two. We rotate the sequence of stripes in $\textsc{Str}_1$ in such a way that K_{ij} becomes the first stripe and rename the stripes into $L_1, \ldots, L_\ell$ with $L_1 = K_{ij}$. In case $C[L_1, I \setminus L_1]$ is a sum-matrix, $V = L_1$ and $W = I \setminus L_1$ form a nice bipartition. Otherwise, we observe that any Kalmanson permutation in $\textsc{Str}(L_1, \ldots, L_\ell)$ must transform $C[L_1, I \setminus L_1]$ into a Contra Monge matrix. According to Proposition 2.2, we compute an appropriate partition of L_1 and an appropriate partition of $I \setminus L_1$ that encodes all permutations that transform $C[L_1, I \setminus L_1]$ into a Contra Monge matrix. This either results in a refinement of the stripes in $L_1, \ldots, L_\ell$ (cf. Proposition 2.1) or the set of potential permutations becomes empty (and C is not a permuted Kalmanson matrix).

This procedure is repeated over and over again as long as there are stripes of cardinality at least two. Either we find a nice bipartition of I, or we may refine the stripes, or eventually all stripes are of cardinality one. Since the stripes can be refined

at most $(n-1)$ times, a conservative estimation yields $D(n) = O(n^3)$. According to the above arguments, the recognition algorithm runs in polynomial time $O(n^4)$.

4.2. Implementation of the algorithm. In this subsection, we explain how to implement the divide and conquer algorithm described in the preceding section in $O(n^2 \log n)$ time. Our main tools are advanced data structures (PQ-trees and union-find structures) and a slight modification of the divide step. Let us start with three simple but important statements.

• All through the algorithm we will derive and exploit sufficient conditions on the matrix for being Kalmanson. These conditions will restrict and cut down the set of potentially feasible permutations. We will not verify at every single step whether we are indeed dealing with Kalmanson permutations (this would be too time consuming). Hence, the output of the algorithm will be some $\sigma \in S_n$ with the following property: "In case C is a permuted Kalmanson matrix, then $C_\sigma \in \mathbb{K}$." (Cf. Lemma 4.1). The verification of whether C_σ is indeed in $\mathbb{K}$ is postponed to a single $O(n^2)$ check in the end, *after* the algorithm.

• All sets of permutations induced by stripes are stored in PQ-trees of height two (as already stated in section 2). For a set of permutations Π over I stored in a PQ-tree of constant height and a subset $J \subseteq I$, the following operation can be performed in $O(|J|)$ time: "Restructure the PQ-tree in such a way that the restructured PQ-tree stores exactly those permutations $\pi \in \Pi$ in which the objects in J are consecutive" (cf. Booth and Lueker [1]). Note that this operation might lead to an empty set of permutations.

• The algorithm represents its knowledge on the equivalence of rows in a union–find data structure in order to answer in constant time questions of the form "Are the rows i and j already known to be equivalent?". In case it receives new information on the equivalence of two rows i and j, the corresponding two equivalence classes have to be combined into a single class. We choose an implementation of this data structure that supports the FIND operation in constant time and that supports the UNION operation in time that is linear in the size of the merged classes (this can be done, for example, via pointers from every element to the name of the corresponding class; see Cormen, Leiserson, and Rivest [3]).

Next, we give a precise low-level description of the algorithm. The algorithm performs the following five steps (S0)–(S4).

(S0) If $n \leq 3$, output any permutation $\sigma \in S_n$.

(S1) Row 1 is compared to rows k, $2 \leq k \leq n$ until some row k is found that is not equivalent to 1 or all rows are known to be equivalent to row 1. If $1 \not\sim k$, the algorithm moves on to (S2). If all rows are found to be equivalent to row 1, the algorithm outputs the identity permutation and stops.

Observing that any $n \times n$ matrix with $n \leq 3$ is a Kalmanson matrix justifies step (S0). In step (S1), a comparison between rows 1 and k is performed as follows: first, the algorithm checks whether row 1 is already known to be equivalent to k (from some higher level of the recursion). In case it is, the algorithm immediately moves on to the next row. Otherwise, it scans row k in linear time. If it turns out that $1 \sim k$, this information is handed over to the union–find structure and the next row is investigated.

The following step (S2) is a kind of initialization for step (S3).

(S2) Let $k \not\sim 1$ be the result of (S1). Compute $\mathcal{M}(1,k)$ and define the sets $K_1 = \mathcal{M}(1,k) \cup \{1\}$ and $K_2 = I \setminus K_1$. Compute for $C[K_1, K_2]$ the partitions $K_{11}, \ldots, K_{1r}$ of K_1 and $K_{21}, \ldots, K_{2s}$ of K_2 that encode all permutations that

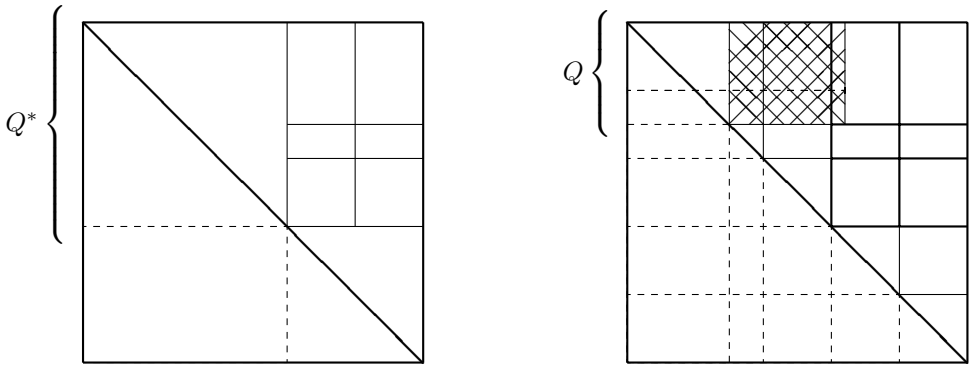


FIG. 4.2. *Illustration for step (S3) of the algorithm: the cross-hatched region is matrix D; the last column of D is q.*

transform $C[K_1, K_2]$ into a Contra Monge matrix (this is done according to Proposition 2.2).

Set $\text{STR}_0 = \text{STR}(K_1, K_2)$ and $\text{STR}_1 = \text{STR}(K_{11}, \ldots, K_{1r}, K_{21}, \ldots, K_{2s})$.

By the discussion in subsection 4.1, it is sufficient to search for Kalmanson permutations in STR_1. Note that from the submatrix $C[K_1, K_2]$, we will not receive any further information on refining the stripes: by Proposition 2.2(iii) for every stripe K_{1i}, the matrix $C[K_{1i}, K_2]$ is a sum-matrix and for every stripe K_{2j}, the matrix $C[K_1, K_{2j}]$ is a sum-matrix. However, we may receive information from $C[K_{1i}, K_1]$ and $C[K_{2j}, K_2]$.

In the following "refinement step" (S3), a sequence of sets of permutations $\text{STR}_1, \text{STR}_2, \ldots$ is constructed with $\text{STR}_i = \text{STR}(Q_1^{(i)}, \ldots, Q_{m_i}^{(i)})$. The partition in STR_{i+1} is always a refinement of the partition in STR_i. The refinement procedure stops as soon as, for every stripe $Q = Q_j^{(i)}$, the submatrix $C[Q, I \setminus Q]$ is a sum-matrix.

(S3) Starting with $j = 1$, perform the following refinement procedure for every stripe $Q = Q_j^{(i)}$ in STR_i: rotate STR_i in such a way that Q becomes the first stripe in STR_i. Let Q^* be the *mother stripe* of Q in STR_{i-1}, i.e., the stripe that contains set Q. Let q be any column that is *not* in Q^*. In case $Q \neq Q^*$ holds, consider the matrix $D = C[Q, (Q^* \setminus Q) \cup \{q\}]$ and compute according to Proposition 2.2 the partition of the rows and columns of D that implicitly describe all permutations that transform D into a Contra Monge matrix. If $Q = Q^*$ holds, consider the next set $Q_j^{(i)}$.

The set STR_{i+1} results from refining STR_i according to all these new partitions. Step (S3) is repeated until $\text{STR}_{i+1} = \text{STR}_i$, i.e., the partition has not been refined.

Consider the matrices $E = [Q, I \setminus Q]$ and $F = [Q, I \setminus Q^*]$ (cf. Figure 4.2). Since Q was obtained as a stripe from Q^*, Proposition 2.2(iii) ensures that the matrix F is a sum-matrix. By Lemma 3.1, we may restrict our attention to permutations that transform matrix E into a Contra Monge matrix. E essentially decomposes into D and F (where D and F have the common column q). Proposition 2.2 states that in Contra Monge matrices, sum–submatrices may as well be represented by a single column (in our case by column q). This simplifies matrix E down to matrix D.

If no more refinement of a stripe Q is possible, it follows from Proposition 2.2 that

the corresponding matrix D is a sum-matrix. Since F is a sum-matrix, too, and the sum-matrices D and F have a common column q, this implies that the matrix E itself is a sum-matrix. Hence at the end of (S3), all matrices $C[Q, I \setminus Q]$ are sum-matrices.

Note that some of the derived refinements may contradict each other: we receive constraints (i.e., subsets of I that must be consecutive) from every single stripe Q. The constraints arise from the rows and from the columns of the Contra Monge matrices and they also concern the other stripes within Q^*. Another type of contradiction arises if the Contra Monge conditions force q to become an interior column of D. Hence, if some constraint cannot be fed into the PQ-tree, there does not exist a consistent refinement for Q^* and matrix C cannot be a permuted Kalmanson matrix. In this case, the algorithm returns any permutation and stops.

(S4) Recursion. Let $\textsc{Str}_i = \textsc{Str}(Q_1^{(i)}, \ldots, Q_m^{(i)})$ be the resulting set of permutations as derived in step (S3). For every stripe $Q_j^{(i)}$, select any row q_{j+1} from $Q_{j+1}^{(i)}$. Set $X_j = Q_j^{(i)} \cup \{q_{j+1}\}$ and determine the restriction of the union-find structure to the elements in X_j. Compute recursively a Kalmanson permutation π_j for $C[X_j, X_j]$ under this union-find structure. Afterwards, rotate π_j such that q_{j+1} becomes the last element and remove q_{j+1} from the rotated permutation. This yields permutation σ_j.
The output consists of the concatenation of the permutations $\sigma_1, \sigma_2, \ldots, \sigma_m$ in exactly this order.

Step (S4) essentially applies Lemma 3.3 to the stripes in $\textsc{Str}_i$.

LEMMA 4.1. *Either matrix C is not a permuted Kalmanson matrix, or $C_\sigma \in$ ⊮ holds for the output permutation $\sigma \in S_n$ of the above algorithm.*

The correctness of this statement follows from the lemmas derived in section 3. It remains to investigate the time complexity of the algorithm. In doing this, it is convenient to make a separate analysis for the main part of the algorithm and a separate analysis for the UNION operations.

We first investigate the main part of the algorithm. Let $\textsc{Str}_i = \textsc{Str}(Q_1^{(i)}, \ldots, Q_m^{(i)})$ be the final set of permutations computed in step (S3) and let a_j denote the cardinality of $Q_j^{(i)}$. Clearly, $a_j \le n-2$ for all j and $\sum_{j=1}^m a_j = n$. Define $S_a = \sum_{j=1}^m a_j^2$. For every j, the algorithm treats the submatrix corresponding to stripe $Q_j^{(i)}$ recursively. The remaining area of size $n^2 - S_a$ is covered by small, almost disjoint submatrices (they are disjoint with the exception of the negligible columns used to represent the sum-submatrices). Such a covering submatrix with dimensions $x \times y$ is handled in $O(xy + x \log x + y \log y)$ time according to Proposition 2.2. Hence, handling a matrix of area A is done in $O(A \log A)$ time and this complexity is superlinear in the concerned area. Thus, the total cost for handling all these matrices is at most $O\big((n^2 - S_a) \log(n^2 - S_a)\big)$. Storing, refining, and modifying the partitions with the help of PQ-trees costs time that is linear in the size of the concerned set (i.e., proportional to the sidelengths of the submatrices) and thus is dominated by the cost for handling the submatrices. The overall cost for the FIND operations in step (S1) is $O(n)$. Summarizing, the time $T(n)$ for treating a matrix of sidelength n obeys

$$T(n) \le \max_{a_j} \left\{ \sum_{j=1}^m T(a_j + 1) + c_1 \left(n^2 - \sum_{j=1}^m a_j^2 \right) \log_2 \left(n^2 - \sum_{j=1}^m a_j^2 \right) + c_2 n \right\}, \tag{4.2}$$

where the maximum is taken over all m-tuples of integers a_j with $1 \le a_j \le n-2$ and $\sum_{j=1}^m a_j = n$, and where c_1 and c_2 are appropriate positive constants.

LEMMA 4.2. $T(n) = O(n^2 \log n)$.

Proof. Define $c_3 = \max\{20c_1, 20c_2, T(2), T(3)\}$. We prove by induction that for all $n \geq 2$, $T(n) \leq c_3(n-1)^2 \log_2 n$ holds. By the definition of c_3, this inequality holds true for $n = 2$ and $n = 3$. Next consider some fixed $n \geq 4$ and consider an m-tuple of integers a_j that maximizes the expression in the right-hand side of (4.2). Observe that $S_a = \sum_{j=1}^{m} a_j^2 \leq n^2 - 4n + 8$ holds, and use the inductive assumption to derive that $\sum_{j=1}^{m} T(a_j + 1) \leq c_3 S_a \log_2 n$. Hence, $T(n) \leq c_3 S_a \log_2 n + c_1(n^2 - S_a)\log_2(n^2) + c_2 n$, and the lemma follows. □

LEMMA 4.3. *The total time needed for performing all UNION operations is bounded by* $O(n^2)$.

Proof. The union-find structure is only used in step (S1). Since the cost for the FIND operations was already investigated above, it remains to analyze the UNION operations. We represent the recursive process in the standard way by a tree: the root of the tree represents the original problem. The sons of a vertex v in the tree represent the subproblems originating in step (S4) when the problem corresponding to v is treated. Every vertex v is labeled by two numbers a_v and b_v, where a_v equals the number of rows (i.e., the *size*) of the corresponding subproblem and b_v denotes the number of UNION operations that result from treating the subproblem. Clearly, the a-label of the root equals n, and $b_v \leq a_v$ holds for every vertex v.

Since every UNION operation decreases the number of equivalence classes by one and since in every leaf of the tree there remains at least one equivalence class, on every branch going from some leaf up to the root, the overall sum of all values b_v is at most $n-1$. Moreover, the sum of the a-labels of all sons of vertex v is bounded by a_v plus the number of sons of v. With this, the total sum of the a-labels of all leaves is $O(n)$. The overall cost of all UNION operations is $O(\sum a_v b_v)$ where the sum is taken over all vertices in the tree. This sum may be bounded from above by another sum that runs over all leaves and adds up the a-label of the leaves times the overall sum of all b-labels on the corresponding path leading from the leaf up to the root. By the above inequalities, this second sum is dominated by $O(n^2)$. This completes the proof of the lemma. □

THEOREM 4.4. *For a symmetric* $n \times n$ *matrix* C*, it can be decided in* $O(n^2 \log n)$ *time whether* C *is a permuted Kalmanson matrix.*

Proof. From Lemma 4.1, we get the correctness and from Lemmata 4.2 and 4.3, we get the time complexity of the algorithm. In the end, we permute C according to the output permutation σ and check whether the permuted matrix C_σ indeed is Kalmanson. This is done in $O(n^2)$ time as described in Proposition 2.4. □

5. Master tours in polynomial time. A *master tour* π for a set V of cities fulfills the following property: for every $V' \subseteq V$, an optimum travelling salesman tour for V' is obtained from π by removing from it the cities that are not in V'. Given the distance matrix C for a set of cities, the *master tour problem* consists in deciding whether this set of cities possesses a master tour. In this section, we prove that the master tour problem is closely related to permuted Kalmanson matrices and hence solvable in polynomial time.

THEOREM 5.1. *For an* $n \times n$ *symmetric distance matrix* C*, the permutation* $\langle 1, 2, \ldots, n\rangle$ *is a master tour if and only if* C *is a Kalmanson matrix.*

Proof. (Only if): Assume that $\langle 1, 2, \ldots, n\rangle$ is a master tour for the distance matrix C. Then by definition, for each subset of four cities $\{i, j, k, \ell\}$ with $1 \leq i < j < k < \ell \leq n$, the tour $\langle i, j, k, \ell\rangle$ is an optimal TSP tour. Since C is symmetric, there are only three combinatorially different tours through those cities: (i) $\langle i, j, k, \ell\rangle$, (ii) $\langle i, j, \ell, k\rangle$

and (iii) $\langle i,k,j,\ell\rangle$. The optimality of tour (i) implies that $c_{ij}+c_{jk}+c_{k\ell}+c_{\ell i}\le c_{ij}+c_{j\ell}+c_{\ell k}+c_{ki}$ and $c_{ij}+c_{jk}+c_{k\ell}+c_{\ell i}\le c_{ik}+c_{kj}+c_{j\ell}+c_{\ell i}$. By exploiting the symmetry of C and simplifying, the above inequalities turn into

$$c_{jk}+c_{i\ell}\le c_{ik}+c_{j\ell} \quad \text{and} \quad c_{ij}+c_{k\ell}\le c_{ik}+c_{j\ell}, \tag{5.1}$$

which are exactly the conditions (1.2) and (1.1). Hence, C is a Kalmanson matrix.

(If): Let $K=\{x_1,\dots,x_k\}$ be a subsequence of $\langle 1,2,\dots,n\rangle$. Then by Proposition 2.5(ii), the matrix $C[K,K]$ is again a Kalmanson matrix and by Kalmanson's result [6] the tour $\langle x_1,\dots,x_k\rangle$ is an optimal tour for K. Consequently, $\langle 1,2,\dots,n\rangle$ is a master tour. □

THEOREM 5.2. *For a symmetric $n\times n$ matrix C, it can be decided in $O(n^2\log n)$ time whether C possesses a master tour.*

Proof. By Theorem 5.1, a symmetric distance matrix has a master tour if and only if it is a permuted Kalmanson matrix. By Theorem 4.4, permuted Kalmanson matrices can be recognized in $O(n^2\log n)$ time. □

6. Discussion. In this paper we have developed an algorithm for recognizing permuted $n\times n$ Kalmanson matrices in $O(n^2\log n)$ time and showed that this problem is equivalent to detecting master tours. Since the input is of size n^2, the derived time complexity is close to optimal. Two questions remain open.

(1) We would like to know whether the $\log n$ factor in the time complexity can be shaved off in the *random access machine* model of computation.

(2) The second question concerns characterizing all Kalmanson permutations for some given input matrix C. Our algorithm just outputs a single Kalmanson permutation. However, we would like to have a complete and concise description of *all* Kalmanson permutation similar to the concise description of all Contra Monge permutations in Proposition 2.2. One of the main obstacles in deriving such a description is that we do not fully understand the structure of equivalent columns. For example, it is *not true* that equivalent columns must stick together in Kalmanson permutations. Consider the following two matrices.

$$A=\begin{pmatrix} * & 0 & 0 & 0\\ 0 & * & 0 & 1\\ 0 & 0 & * & 0\\ 0 & 1 & 0 & * \end{pmatrix} \qquad A_\sigma=\begin{pmatrix} * & 0 & 0 & 0\\ 0 & * & 0 & 0\\ 0 & 0 & * & 1\\ 0 & 0 & 1 & * \end{pmatrix}$$

Matrix A is a Kalmanson matrix where rows 1 and 3 are equivalent. However, its permutation A_σ is not a Kalmanson matrix and it is easy to check that *no* permutation of A which makes rows 1 and 3 neighboring rows yields a Kalmanson matrix.

Note added in proof. In a recent paper presented at the 1996 European Symposium on Algorithms, Christopher, Farach, and Trick answered both of the above questions in the positive. They derived an $O(n^2)$ recognition algorithm and a simple characterization of the set of all Kalmanson permutations that is based on PQ-trees.

Acknowledgment. We would like to thank Bettina Klinz for a careful reading of the paper and for many helpful comments.

REFERENCES

[1] K. S. BOOTH AND G. S. LUEKER, *Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms*, J. Comput. System Sci., 13 (1976), pp. 335–379.

[2] R. E. Burkard, B. Klinz, and R. Rudolf, *Perspectives of Monge properties in optimization*, Discrete Appl. Math., 70 (1996), pp. 95–161.
[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.
[4] V. G. Deĭneko and V. L. Filonenko, *On the Reconstruction of Specially Structured Matrices*, Aktualnyje Problemy EVM i programmirovanije, Dnepropetrovsk, DGU, 1979 (in Russian).
[5] P. C. Gilmore, E. L. Lawler, and D. B. Shmoys, *Well-solved special cases*, in The Travelling Salesman Problem, John Wiley, Chichester, 1985, Chapter 4, pp. 87–143.
[6] K. Kalmanson, *Edgeconvex circuits and the travelling salesman problem*, Canad. J. Math., 27 (1975), pp. 1000–1010.
[7] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *The Travelling Salesman Problem*, John Wiley, Chichester, 1985.
[8] C. H. Papadimitriou, *Lecture on computational complexity at the Maastricht Summer School on Combinatorial Optimization*, Maastricht, the Netherlands, August, 1993.
[9] C. H. Papadimitriou, *Computational Complexity*, Addison–Wesley, Reading, MA, 1994.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 94–109, February 1998

007

ANALYSIS OF ALGORITHMS FOR LISTING EQUIVALENCE CLASSES OF k-ARY STRINGS*

ANDRZEJ PROSKUROWSKI†, FRANK RUSKEY‡, AND MALCOLM SMITH‡

Abstract. We give efficient algorithms for listing equivalence classes of k-ary strings under reversal and permutation of alphabet symbols. As representative of each equivalence class, we choose that string which is lexicographically smallest. These algorithms use space $O(n)$ and time $O(\sqrt{k}N)$, where N is the total number of strings generated and n is the length of each string. For $k = 2$, we obtain a recursive decomposition of the set of binary strings that allows the strings to be generated without rejecting any strings. For $k \geq 3$, some strings must be rejected. The algorithm is simple but its exact analysis is rather complicated. In the analysis we determine a quantity of independent interest—the average length of the common prefix of two randomly chosen infinite length "restricted-growth" strings.

Key words. combinatorial generation, k-ary strings, Stirling numbers, restricted growth function, k-paths

AMS subject classifications. 05A15, 05C30, 60J10, 68Q25, 68R05, 68R15

PII. S0895480192234009

1. Introduction.

What are the most natural group actions on strings over a fixed alphabet? Four actions immediately suggest themselves: (a) leaving the string unchanged, (b) reversing a string, (c) rotating a string, and (d) permuting the symbols of the string by a permutation of the alphabet. The four groups giving rise to these actions are (a) $\mathbb{Z}_1$, (b) $\mathbb{Z}_2$, (c) the cyclic group $\mathbb{C}_n$, and (d) the symmetric group $\mathbb{S}_k$, assuming the alphabet consists of k symbols.

Each group action, or composition of group actions, partitions the set of k-ary strings into equivalence classes, namely the orbits of the action. To generate these equivalence classes, it is natural to choose as representative the lexicographically smallest string. With this representation, efficient algorithms are known for generating the equivalence classes of each of the actions (a), (b), (c), and (d). By "efficient" we mean that the amount of computation used in generating the objects is proportional to the number of objects generated. For (a) we are simply counting in base k which is known to be efficient for $k \geq 2$. For (b), efficient algorithms were developed by Ruskey [15]. In case (c) the equivalence classes are usually called *necklaces*. Efficient algorithms for generating necklaces were developed by Fredricksen and Kessler [2] and Fredricksen and Maiorana [3]; these algorithms were proven to be efficient by Ruskey, Savage, and Wang [14]. In case (d) the representative strings are usually called *restricted growth functions* and efficient algorithms for generating them have been developed by Er [1], Kaye [8], and others.

In contrast to the case where our three nontrivial actions are considered in isolation, the composition of more than two of the actions gives rise to equivalence classes

*Received by the editors July 6, 1992; accepted for publication (in revised form) February 20, 1997.

http://www.siam.org/journals/sidma/11-1/23400.html

†Department of Computer and Information Science, University of Oregon, Eugene, OR 97403 (andrzej@cs.uoregon.edu). The research of this author was supported in part by NSF grant CCR-9213431.

‡Department of Computer Science, University of Victoria, Victoria, B.C., V8W 3P6, Canada (fruskey@csr.uvic.ca, smith@CamosunBC.CA). The research of F. Ruskey was supported in part by NSERC grant OGP0003379.

for which no efficient generation algorithms were previously known. For example, composing (b) and (c) results in the dihedral group, with the resulting equivalence classes known as *bracelets*. No more efficient algorithm is known than simply listing all necklaces and rejecting those that are larger than their reversals (i.e., are not representative of a bracelet). In this paper we compose (b) and (d) and develop efficient (for fixed k) algorithms for generating the resulting equivalence classes. It remains an interesting challenge to develop efficient algorithms for the other compositions.

Let us recast our problem. The problem is to list equivalence classes of k-ary strings under the transitive closure of the binary relation $\mathcal{R}$ defined on k-ary strings of length n by

$$x_1 \cdots x_n \ \mathcal{R} \ y_1 \cdots y_n \Leftrightarrow \begin{cases} x_1 \cdots x_n = y_n \cdots y_1 \text{ or} \\ x_1 \cdots x_n = \pi(y_1) \cdots \pi(y_n) \text{ for some } \pi \in \mathbb{S}_k. \end{cases} \tag{1.1}$$

As representative of each equivalence class we choose the lexicographically smallest string.

An important tool in listing sets of combinatorial objects consists of recurrence relations describing the sets. The typical operation on lists is *concatenation* ($\oplus$) corresponding to *addition* ($+$) of cardinalities and to *union* of disjoint sets, ($\uplus$). Strings are denoted using lowercase bold letters. Sets and lists of strings are indicated by uppercase bold letters, with their corresponding cardinalities indicated by uppercase (nonbold) roman letters. Thus, as a trivial example, the set of all binary strings of length n will be denoted $\mathbf{B}_n$ fulfilling the recurrence relation

$$\mathbf{B}_n = 0\mathbf{B}_{n-1} \uplus 1\mathbf{B}_{n-1} \qquad \text{for } n > 0, \tag{1.2}$$

where $\mathbf{B}_0 = \{\epsilon\}$; the corresponding recurrence relation for its cardinality is $B_n = 2B_{n-1}$ with $B_0 = 1$, which has the closed form solution $B_n = 2^n$. Additionally, strings can be *reversed*. The reversal of a string $\mathbf{s}$ is denoted $\mathbf{s}^R$; this notation is extended to sets of strings in the natural manner. For example, $\{0011, 0101\}^R = \{1100, 1010\}$. If $\mathbf{s}$ and $\mathbf{t}$ are strings then by $\mathbf{s} \leq \mathbf{t}$ we mean that $\mathbf{s}$ is lexicographically less than or equal to $\mathbf{t}$; comparisons among strings are always made with respect to lexicographic order.

An algorithm for generating combinatorial objects is said to be CAT, standing for *Constant Amortized Time*, if it has running time $O(p(n) + N)$, where N is the total number of objects generated, and $p(n)$ is a polynomial in n, where n is the "size" of the object being generated. The term $p(n)$ is meant to represent any time spent in preprocessing. In this paper N is exponential in n and so the $p(n)$ term can be ignored. The algorithms that we consider are recursive and the underlying computation tree provides a great conceptual aid in analyzing the behavior of the algorithm. In many algorithms the total amount of computation is proportional to the number of nodes in the computation tree; the algorithms we present have this feature. A desirable property of a generation algorithm is that an object is produced at every leaf (terminal vertex) of the computation tree. We call this the BEST[1] property (for Backtracking Ensuring Success at Terminals). The *CAT principle* states sufficient conditions for a BEST algorithm to run in constant amortized time. These conditions are that (a) in the computation tree, there are not "too many" nodes in the computation tree with a single child, and (b) the total computation can be partitioned so that each node is assigned constant time computation. In particular, if the degree of each nonleaf is at

[1]The CAT and BEST acronyms are due to the second author.

least two, then there are more leaves than internal nodes and so condition (a) is met. The CAT principle is the most common way of showing that a recursive algorithm runs in constant amortized time.

In section 2 we present a BEST CAT algorithm for the binary case $k = 2$. In section 3 we present a nonBEST algorithm that is CAT for fixed $k \geq 2$; this algorithm is analyzed in sections 4 and 5. The analysis of section 4 provides an exact count of the number of operations used by the algorithm for $k = 3$ and $k = 4$. In principle this type of analysis could be extended to larger values of k. The asymptotic analysis of section 5 is based on the expected length of the longest common prefix of two infinite canonical k-ary strings. This quantity, which turns out to be $O(\sqrt{k})$, is of independent mathematical interest. In section 6 we indicate how our algorithms solve the problem of listing nonisomorphic k-paths, which was our initial motivation for studying the problem. The final section suggest some interesting open problems.

2. Generating binary strings. In order to analyze equivalence classes under the group actions of symbol permutations (i.e., complementation in the binary case) and under string reversal, we shall look first at the problem of listing all equivalence classes under reversal only. That is to say, two bitstrings $a_1a_2\cdots a_n \neq b_1b_2\cdots b_n$ are considered equivalent if $a_1a_2\cdots a_n = b_n\cdots b_2b_1$. Given two equivalent bitstrings we will list the one that is lexicographically (i.e., numerically) smaller. Thus, for example, 1010011 is listed but 1100101 is not. Let the set of listed bitstrings be denoted $\mathbf{M}_n$.

$$\mathbf{M}_n \stackrel{\text{def}}{=} \{\mathbf{s} \in \mathbf{B}_n \mid \mathbf{s} \leq \mathbf{s}^R\}. \tag{2.1}$$

The recurrence relation for this set follows the observation that only identical first and last bits allow for symmetry under reversal. Thus, $\mathbf{M}_0 = \{\epsilon\}$, $\mathbf{M}_1 = \{0, 1\}$, and

$$\mathbf{M}_n \;=\; 0\mathbf{M}_{n-2}0 \;\uplus\; 0\mathbf{B}_{n-2}1 \;\uplus\; 1\mathbf{M}_{n-2}1. \tag{2.2}$$

We use $\mathbf{L}_n$ to denote those elements of $\mathbf{M}_n$ that are nonpalindromic.

$$\mathbf{L}_n \stackrel{\text{def}}{=} \{\mathbf{s} \in \mathbf{B}_n \mid \mathbf{s} < \mathbf{s}^R\} \;=\; \{\mathbf{s} \in \mathbf{M}_n \mid \mathbf{s} \neq \mathbf{s}^R\}.$$

These sets of strings satisfy a recurrence relation of the same form as (2.2), but with different initial conditions. Here, $\mathbf{L}_0 = \mathbf{L}_1 = \emptyset$. The recurrence relation is

$$\mathbf{L}_n \;=\; 0\mathbf{L}_{n-2}0 \;\uplus\; 0\mathbf{B}_{n-2}1 \;\uplus\; 1\mathbf{L}_{n-2}1. \tag{2.3}$$

If $\mathbf{s}$ is a bitstring then by $\overline{\mathbf{s}}$ we denote the complement of $\mathbf{s}$. For example, $\overline{1010011} = 0101100$.

OBSERVATION 1. *The operations of complement and reversal commute. That is, for all binary strings* $\mathbf{s}$*, we have* $\overline{\mathbf{s}^R} = \overline{\mathbf{s}}^R$.

OBSERVATION 2. *For all binary strings* $\mathbf{s}$ *and* $\mathbf{t}$*, we have* $\mathbf{s} \leq \mathbf{t}$ *if and only if* $\overline{\mathbf{s}} \geq \overline{\mathbf{t}}$.

OBSERVATION 3. *For all binary strings, if* $\mathbf{s} \leq \mathbf{s}^R$ *and* $\mathbf{s} \leq \overline{\mathbf{s}}^R$*, then* $\mathbf{s} \leq \overline{\mathbf{s}}$.[2]

Observations 1 and 2 are trivial. To prove Observation 3, note that by Observation 2, if $\mathbf{s} \leq \mathbf{s}^R$, then $\overline{\mathbf{s}} \geq \overline{\mathbf{s}}^R$. Thus, $\overline{\mathbf{s}} \geq \overline{\mathbf{s}}^R \geq \mathbf{s}$.

Let $\mathbf{Y}_n$ denote the set of all bitstrings of length n that are lexicographically smallest in the equivalence classes induced by relation $\mathcal{R}$, which is defined by (1.1)

[2]Also, $\mathbf{s} \leq \overline{\mathbf{s}}$ and $\mathbf{s} \leq \mathbf{s}^R$ implies that $\mathbf{s} \leq \overline{\mathbf{s}}^R$. However, the example of $\mathbf{s} = 0100$ shows that $\mathbf{s} \leq \overline{\mathbf{s}}$ and $\mathbf{s} \leq \overline{\mathbf{s}}^R$ do not imply $\mathbf{s} \leq \mathbf{s}^R$.

```
procedure Y ( lo, hi : integer );
begin
  if lo = hi then begin
     s[lo] := 0;  {s[hi] := 0;}  PrintIt;
  end else
  if lo+1 = hi then begin
     s[lo] := 0;   s[hi] := 0;   PrintIt;
     s[lo] := 0;   s[hi] := 1;   PrintIt;
  end else begin
     s[lo] := 0;   s[hi] := 0;   Mr( lo+1, hi-1 );
    {s[lo] := 0;}  s[hi] := 1;   Nr( lo+1, hi-1 );
    {s[lo] := 0;} {s[hi] := 1;}   Y( lo+1, hi-1 );
  end
end {of Y};
```

FIG. 2.1. *Pascal code for generating* $\mathbf{Y}_n$.

in the introduction. Let $\mathbf{N}_n$ be the set of all nonpalindromic elements of $\mathbf{Y}_n$. For example, $\mathbf{N}_4 = \{0010, 0001, 0011, 0101\}$ and $\mathbf{Y}_4 = \mathbf{N}_4 \cup \{0000, 0110\}$. More formally,

$$\mathbf{Y}_n \stackrel{\text{def}}{=} \{\mathbf{s} \in \mathbf{B}_n \mid \mathbf{s} \le \overline{\mathbf{s}}, \mathbf{s} \le \mathbf{s}^R, \text{ and } \mathbf{s} \le \overline{\mathbf{s}}^R\} \tag{2.4}$$

and

$$\mathbf{N}_n \stackrel{\text{def}}{=} \{\mathbf{s} \in \mathbf{Y}_n \mid \mathbf{s} \ne \mathbf{s}^R\}. \tag{2.5}$$

The sets $\mathbf{Y}_n$ and $\mathbf{N}_n$ admit the following recursive decompositions.

LEMMA 2.1. *For all* $n > 1$,

$$\mathbf{Y}_n = 0\mathbf{Y}_{n-2}1 \uplus 0\mathbf{N}_{n-2}^R 1 \uplus 0\mathbf{M}_{n-2}0 \tag{2.6}$$

and

$$\mathbf{N}_n = 0\mathbf{Y}_{n-2}1 \uplus 0\mathbf{N}_{n-2}^R 1 \uplus 0\mathbf{L}_{n-2}0. \tag{2.7}$$

The initial values are $\mathbf{Y}_0 = \{\epsilon\}$, $\mathbf{Y}_1 = \{0\}$, *and* $\mathbf{N}_0 = \mathbf{N}_1 = \emptyset$.

Proof. To prove (2.6), first note that a string $\mathbf{s}$ in $\mathbf{Y}_n$ cannot start with a 1. If $\mathbf{s} = 0\mathbf{t}0$, then $\mathbf{s} \le \overline{\mathbf{s}}$ and $\mathbf{s} \le \overline{\mathbf{s}}^R$, so the only remaining condition that has to be satisfied is $\mathbf{s} \le \mathbf{s}^R$. This is true if and only if $\mathbf{t} \le \mathbf{t}^R$. Thus, a string of the form $0\mathbf{t}0$ is in $\mathbf{Y}_n$ if and only if $\mathbf{t} \in \mathbf{M}_{n-2}$. If $\mathbf{s} = 0\mathbf{t}1$, then $\mathbf{s} \le \mathbf{s}^R$ and $\mathbf{s} \le \overline{\mathbf{s}}$, so the only remaining condition that has to be satisfied is $\mathbf{s} \le \overline{\mathbf{s}}^R$. By Observation 3, $\mathbf{Y}_n = \{\mathbf{s} \in \mathbf{B}_n \mid \mathbf{s} \le \mathbf{s}^R, \mathbf{s} \le \overline{\mathbf{s}}^R\}$. By Observation 2, the condition $\mathbf{s} \le \overline{\mathbf{s}}^R$ is equivalent to the condition $\mathbf{s}^R \le \overline{\mathbf{s}}$. Thus,

$$\begin{aligned}\{\mathbf{s} \in \mathbf{B}_n \mid \mathbf{s} \le \overline{\mathbf{s}}^R\} &= \{\mathbf{s} \in \mathbf{B}_n \mid (\mathbf{s} \le \mathbf{s}^R \text{ and } \mathbf{s} \le \overline{\mathbf{s}}^R) \text{ or } (\mathbf{s}^R \le \mathbf{s} \text{ and } \mathbf{s}^R \le \overline{\mathbf{s}})\} \\ &= \mathbf{Y}_n \cup \mathbf{Y}_n^R.\end{aligned}$$

Hence, a string of the form $0\mathbf{t}1$ is in $\mathbf{Y}_n$ if and only if $\mathbf{t} \in \mathbf{Y}_{n-2} \cup \mathbf{Y}_{n-2}^R$. Recurrence relation (2.6) follows by partitioning $\mathbf{Y}_{n-2} \cup \mathbf{Y}_{n-2}^R$ into the two disjoint sets $\mathbf{Y}_{n-2}$ and $\mathbf{N}_{n-2}^R$.

The proof of (2.7) is similar and is omitted. □

These equations allow us to write a program for listing the elements of $\mathbf{Y}_n$ that runs in constant amortized time. One could write separate procedures for listing the elements of $\mathbf{Y}_n$, $\mathbf{N}_n$, $\mathbf{L}_n$, $\mathbf{M}_n$, and $\mathbf{B}_n$. Pascal code for $\mathbf{Y}_n$ is given in Figure 2.1.

The array `s[1··n]` of bits is global. Each procedure has two parameters `lo` and `hi` indicating the subarray of `s` whose bits still remain to be set. The initial call is `Y(1,n)`, with no initialization necessary. Note that, aside from the recursive calls, only a constant amount of computation is done within `Y`. Furthermore, any nonleaf call to `Y` has at least two (in fact three) children in the computation tree. Since each of the recurrences (1.2), (2.2), (2.3), (2.6), and (2.7) has at least two terms, these same observations apply to other procedures called by `Y`. Thus we can invoke the CAT principle to conclude that the algorithm runs in constant amortized time.

3. Generating k-ary strings for $k > 2$. In this section, we develop an algorithm for generating the lexicographically smallest k-ary strings from each equivalence class induced by the actions of reversing a string and permuting the symbols of its alphabet. We first generate the lexicographically smallest k-ary strings from each equivalence class induced only by the action of permuting alphabet symbols, and then reject those strings that are not also lexicographically smallest when both group actions are used.

DEFINITION 3.1. *If* $\mathbf{s}$ *is a* k*-ary string, then by* $can(\mathbf{s})$ *we denote the lexicographically smallest string* $\pi(\mathbf{s})$*, taken over all permutations* π *of* $\{0, 1, \ldots, k-1\}$*. If* $\mathbf{s} = can(\mathbf{s})$*, then we say that* $\mathbf{s}$ *is* canonical*. The set of canonical* k*-ary strings of length* n *is denoted* $\mathbf{X}_{n,k}$*.*

For example, $can(660240032644) = 001231142033$. There may be several permutations π for which $\pi(\mathbf{s}) = can(\mathbf{s})$ (we think of the permutation π as an element of a group acting on the string $\mathbf{s}$). In the example above, $\pi = (0\ 1\ 6)(2)(3\ 4)(5)$ and $\pi = (0\ 1\ 5\ 6)(2)(3\ 4)$ are such permutations. In order to define a unique permutation $\pi_{\mathbf{s}}$ such that $\pi_{\mathbf{s}}(\mathbf{s}) = can(\mathbf{s})$, we assume that all elements of $0, 1, \ldots, k-1$ not used in $\mathbf{s}$ or in $can(\mathbf{s})$ form 1-cycles in $\pi_{\mathbf{s}}$. Thus $\pi_{\mathbf{s}} = (0\ 1\ 6)(2)(3\ 4)(5)$ in our example. We will write permutations in cycle notation because it is the cycle structure of these permutations that will be of most importance to us in the ensuing discussion.

The following lemma, whose proof is immediate, characterizes canonical strings.

LEMMA 3.2. *A string* $s = s_1 s_2 \cdots s_n$ *is canonical if and only if* $s_1 = 0$ *and for all* $i = 1, \ldots, n-1$*,*

$$s_{i+1} \ \leq \ \min(k-1, 1 + \max\{s_1, \ldots, s_i\}). \tag{3.1}$$

Sequences satisfying the conditions of Lemma 3.2 are often called "restricted-growth" functions (see Stanton and White [17]) and are studied in connection with set partitions. The Stirling numbers of the second kind, denoted $\left\{ {n \atop k} \right\}$, count the number of restricted growth functions of length n with maximal value k. Let $X_{n,k} = |\mathbf{X}_{n,k}|$. It is well known that

$$X_{n,k} = \sum_{i=1}^{k} \left\{ {n \atop i} \right\}. \tag{3.2}$$

What is the asymptotic value of $X_{n,k}$ for fixed k and large n? Note that $X_{n,k}$ is the number of ways of placing n labeled balls into k unlabeled boxes and thus that $k^n/k! \leq X_{n,k}$ since there are at most $k!$ distinct ways of labeling the boxes once the balls have been placed. On the other hand, $\left\{ {n \atop i} \right\} \leq i^n/i!$ since $i!\left\{ {n \atop i} \right\}$ is the number of surjective functions from an n-set onto an i-set. Thus

$$\frac{k^n}{k!} \ \leq \ X_{n,k} \ \leq \ \sum_{i=1}^{k} \frac{i^n}{i!}.$$

From these upper and lower bounds we obtain the following asymptotic expression for $X_{n,k}$ (with k fixed and n large).

$$X_{n,k} \sim \frac{k^n}{k!}. \tag{3.3}$$

DEFINITION 3.3. *If* $\mathbf{s}$ *is a canonical* k*-ary string, then by* $mate(\mathbf{s})$ *we denote the string* $can(\mathbf{s}^R)$. *If* $\mathbf{s} = mate(\mathbf{s})$, *then we say that* $\mathbf{s}$ *is* symmetric. *Let* $c(\mathbf{s})$ *denote the length of the longest prefix common to both of* $\mathbf{s}$ *and* $mate(\mathbf{s})$.

Since the first symbol of both $\mathbf{s}$ and $mate(\mathbf{s})$ is a zero, $1 \le c(\mathbf{s}) \le n$, where n is the length of $\mathbf{s}$. Continuing our example, if $\mathbf{s} = 001231142033$, then $mate(\mathbf{s}) = can(330241132100) = 001234402411$ and $c(\mathbf{s}) = 5$ since the largest prefix on which they agree is 00123. An example of a symmetric string is 0010231011.

We define $\mathbf{Y}_{n,k} \stackrel{\text{def}}{=} \{\mathbf{s} \in \mathbf{X}_{n,k} \mid \mathbf{s} \le mate(\mathbf{s})\}$.

Lemma 3.2 gives rise to a simple-minded algorithm generating all canonical strings of length n. The algorithm, which we call `gen`, is given in Figure 3.1. A similar algorithm for generating all set partitions (i.e., when the maximum block size k is n) was given by Er [1], and other iterative algorithms are given by Hutchinson [7], Kaye [8], Semba [16], and Stanton and White [17].

```
procedure gen( l, m : integer );
var i : integer;
begin
   if l > n (* and Check *) then PrintIt
   else begin
       for i := 0 to m do begin
          s[l] := i;
          gen( l+1, m );
       end;
       if m < k-1 then begin
          s[l] := m+1;
          gen( l+1, m+1 );
       end;
   end;
end {of gen};
```

FIG. 3.1. *Pascal procedure to generate* $\mathbf{X}_{n,k}$ *(and* $\mathbf{Y}_{n,k}$*).*

The call `gen(l,m)` generates all sequences $s[l], \ldots, s[n]$ such that $0 \le s[i] \le \min(k-1, 1+\max\{m, s[l], \ldots, s[i-1]\})$ for $i = l, l+1, \ldots, n$. Here, parameter `m` is the maximum value of `s[1..l-1]`. Given n, the initial call is `gen(1,0)`. Assuming $k > 1$, note that every call to `gen(l,m)` for which $l \le n$ generates at least two calls to `gen(l+1,i)` for some value of i. Thus there are more leaves than internal nodes in the computation tree and so the CAT principle implies that the algorithm runs in constant amortized time. Observe that the algorithm is BEST.

In order to generate the elements of $\mathbf{Y}_{n,k}$ we use `gen` and check whether each generated string $\mathbf{s}$ satisfies $\mathbf{s} \le mate(\mathbf{s})$. This checking is done by the procedure `Check` of Figure 3.2. With the addition of `Check`, algorithm `gen` is no longer BEST. Clearly, the amount of computation done by `Check`$(\mathbf{s})$ is $O(k+c(\mathbf{s}))$. The term k comes from the initialization (marked `#1` in Figure 3.2), but that term can be eliminated by using a constant-time array initialization as explained, for example, in Lewis and Denenberg [10]. With this modification the running time of `Check` is $O(c(\mathbf{s}))$.

To find the amortized cost per string generated we need to determine the average value of $c(\mathbf{s})$, taken over all strings $\mathbf{s}$ in $\mathbf{X}_{n,k}$. Denoting by $X^{(c)}_{n,k}$ the number of strings

```
function Check : boolean;
var
  i, last: integer;
  pi : array[0..100] of integer;
begin
  for i := 0 to k - 1 do pi[i] := maxint; {#1}
  last := 0;
  i := n;
  pi[s[n]] := 0;
  while (i >= 1) and (s[n-i+1] = pi[s[i]]) do begin
     i := i - 1;
     if pi[s[i]] > last then begin
        last := last + 1;
        pi[s[i]] := last;
     end;
  end;
  Check := (i = 0) or (s[n-i+1] < pi[s[i]]);
end {of Check};
```

FIG. 3.2. *The Pascal function* `Check`.

$\mathbf{s} \in \mathbf{X}_{n,k}$ for which $c(\mathbf{s}) = c$, we define the weighted sum

$$S_{n,k} \stackrel{\text{def}}{=} \sum_{c=1}^{n} c \cdot X_{n,k}^{(c)}. \tag{3.4}$$

Define

$$A_{n,k} \stackrel{\text{def}}{=} \frac{S_{n,k}}{X_{n,k}}. \tag{3.5}$$

The quantity of interest is $S_{n,k}/Y_{n,k}$, which is at most $2A_{n,k}$.

4. Exact analysis for small values of k. In this section we give a general classification scheme useful in deriving exact expressions for $S_{n,k}$. Given a canonical string $\mathbf{s}$, this classification is based upon the value of $c(\mathbf{s})$ relative to the length of $\mathbf{s}$ and the cycle structure of $\pi_{\mathbf{s}}$. This scheme is then used to derive exact expressions for $S_{n,3}$ and $S_{n,4}$. The details of these derivations may be found in [12].

We wish to count those strings $\mathbf{s} \in \mathbf{Y}_{n,k}$ with a fixed value of $c(\mathbf{s})$. Let us first make a couple of observations. Consider a fixed canonical string $\mathbf{s}$ and let $c = c(\mathbf{s})$ and $\pi = \pi_{\mathbf{s}^R}$. If $c < n$, then let $x = s_{c+1}$ and $y = s_{n-c}$. Clearly, $\pi(y) \neq x$. There are five basic cases to consider: [I] $1 \leq c < (n-1)/2$; [II] n is odd and $c = (n-1)/2$; [III] n is even and $c = n/2$; [IV] $n/2 < c < n$; and [V] $c = n$.

α	x	β	y	γ

[I]: In this case x is to the left of y in $\mathbf{s}$ as illustrated above. We denote by α the string $s_1 s_2 \cdots s_c$, $\beta = s_{c+2} \cdots s_{n-c-1}$, and $\gamma = s_{n-c+1} \cdots s_n$.

α	x	γ

[II]: In this case $x = y$ as illustrated above. Clearly, we must have $\pi(x) \neq x$. Here $\alpha = s_1 s_2 \cdots s_m$ and $\gamma = s_{m+2} \cdots s_n$, where $n = 2m + 1$.

[III]: In cases III and IV, x is to the right of y in $\mathbf{s}$. We must have $\pi(x) = y$ (and $\pi(y) \neq x$). Thus there is a j-cycle, for some $j \geq 3$, in π of the form

$(\ldots\ x\ \pi(x)\ \pi(\pi(x))\ \ldots) = (\ldots\ x\ y\ z\ \ldots)$ for some z distinct from x and y. Hence, we must have $k \geq 3$ for case III or IV to occur.

α	y	β	x	γ

[IV]: This case is illustrated above. Let β denote the string of symbols between x and y, $\beta = s_{n-c} \cdots s_c$. If $a = s_j$ and $b = s_{n-j+1}$ for some $n - c < j < c$, then we must have $\pi(a) = b$ and $\pi(b) = a$. Thus both a and b are either in 1-cycles $(a)(b)$ or a 2-cycle $(a\ b)$ of π. Hence, in order for case IV to occur (i.e., β is nonempty), we must have $k \geq 4$.

[V]: In case V the string is symmetric and π must consist of 1-cycles and 2-cycles.

4.1. Ternary strings. Here case [IV] cannot occur. For $k = 3$ and even n,

$$S_{n,3} = \frac{5}{16}3^n + \frac{1}{6}(2n-3)3^{n/2} + \frac{n}{4} + \frac{3}{16}.$$

For odd n,

$$S_{n,3} = \frac{5}{16}3^n + \frac{1}{2}(n-2)3^{(n-1)/2} + \frac{n-1}{4} + \frac{9}{16}.$$

The above expressions, together with (3.3) for $k = 3$ show that the average value of $c(\mathbf{s})$ tends to $3! \cdot 5/16 = 15/8 = 1.875$ as n gets large.

4.2. Quaternary strings. We now show the results for $k = 4$. If $n = 2m$, then

$$S_{n,4} = \frac{26}{315}4^n + \frac{5}{12}m2^n + \frac{5}{18}2^n - \frac{15}{14}2^m + \frac{4}{3}m + \frac{32}{45}. \tag{4.1}$$

If $n = 2m + 1$ is odd then we have

$$S_{n,4} = \frac{26}{315}4^n + \frac{1}{3}m2^n + \frac{7}{18}2^n - \frac{9}{7}2^m + \frac{4}{3}m + \frac{53}{45}. \tag{4.2}$$

Thus, when $k = 4$, the average value of $c(\mathbf{s})$ tends to $24 \cdot 26/315 = 1.980952381\ldots$ as n gets large.

In the next section we will show that asymptotically $A_{n,k}$ is $O(\sqrt{k})$ as $n \to \infty$.

5. Asymptotic analysis. For fixed k, we now show how to determine the asymptotic value of $A_{n,k}$. We first show that, asymptotically, the average value of $c(\mathbf{s})$ is the same as the expected position of the first mismatch between two canonic infinite length k-ary strings.

LEMMA 5.1. *Let* $\mathbf{s}$ *be chosen uniformly at random from* $\mathbf{X}_{n,k}$. *With probability tending to* 1 *as* n *increases,* $c(\mathbf{s}) < n/2$.

Proof. If the mismatch occurs at a position greater than $n/2$, then the reversal of the last $n/2$ symbols of $\mathbf{s}$ is equivalent, under some permutation of the k alphabet symbols, to the first $n/2$ symbols of $\mathbf{s}$. The number of such strings is thus at most $k!X_{n/2,k}$. By (3.3), the fraction of these strings in the set $\mathbf{X}_{n,k}$ is $k! \cdot k^{-n/2}$, which tends to 0 when n grows. □

Denote by $c(\mathbf{s}, \mathbf{t})$ the length of the longest prefix common to both $\mathbf{s}$ and $\mathbf{t}$. The previously used notation $c(\mathbf{s})$ is the same as $c(\mathbf{s}, mate(\mathbf{s}))$.

LEMMA 5.2. *Let* $\mathbf{s}$ *be chosen uniformly at random from* $\mathbf{X}_{n,k}$ *and* $\mathbf{t}'$ *be chosen uniformly at random from* $\mathbf{B}_{n,k}$. *Let* $\mathbf{t} = can(\mathbf{t}')$. *Then*

$$\lim_{n\to\infty} A_{n,k} = \sum_{i\geq 1} i \cdot Prob(c(\mathbf{s},\mathbf{t}) = i).$$

Proof. The fraction of strings in $\mathbf{X}_{n/2,k}$ not using symbol k is $X_{n/2,k-1}/X_{n/2,k}$. By (3.3), this is asymptotically $k((k-1)/k)^{n/2}$, which tends to 0 as n grows. Thus, with probability tending to 1, every symbol $0, 1, \ldots, k-1$ occurs among the first $n/2$ symbols of a string $\mathbf{s}$ chosen uniformly at random from $\mathbf{X}_{n,k}$. Thus, asymptotically, the substring $s_{n/2}\cdots s_n$ of $\mathbf{s}$ is a random string in $\mathbf{B}_{n/2,k}$. By the previous lemma, $c(\mathbf{s}) < n/2$ with probability tending to 1. Therefore, comparing $\mathbf{s}$ with its mate is asymptotically equivalent to comparing it with any random string in $\mathbf{B}_{n,k}$. □

It proves useful to have a notation $X_{n,k,p}$ for the number of strings $\mathbf{s} = s_1 s_2 \cdots s_n$ satisfying $s_1 = p$ and the restricted growth condition (3.1). Thus $X_{n,k,0} = X_{n,k}$. These numbers have also been studied before in connection with ranking algorithms for restricted growth functions in lexicographic order (see Williamson [19], [20] where they are called "restricted tail coefficients"). These numbers satisfy the following recurrence relation for $n > 0$.

$$X_{n,k,p} \;=\; (p+1)X_{n-1,k,p} + X_{n-1,k,p+1}. \tag{5.1}$$

LEMMA 5.3. *The following two limits hold.*

$$\lim_{n\to\infty} \frac{X_{n,k,p}}{X_{n+1,k,p}} = \frac{1}{k}, \tag{5.2}$$

$$\lim_{n\to\infty} \frac{X_{n,k,p+1}}{X_{n+1,k,p}} = 1 - \frac{p+1}{k}. \tag{5.3}$$

Proof. Recall that by (3.3) $X_{n,k} \sim k^n/k!$. The lemma follows from the following asymptotic, which may be proven by induction using (5.1).

$$X_{n,k,p} \;\sim\; \frac{(k)_{p+1}k^{n-1}}{k!}.$$

Note. By $(k)_j$ we denote the falling factorial power $(k)_j = k(k-1)\cdots(k-j+1)$. □

LEMMA 5.4. *Let* $\mathbf{s}$ *be an element - of* $\mathbf{X}_{n,k}$ *chosen uniformly at random. Then*

$$\lim_{n\to\infty} Prob_n(s_i = j \mid \max(s_1,\ldots,s_{i-1}) = p) \;=\; \begin{cases} 1/k & \text{if } 0 \leq j \leq p, \\ 1-(p+1)/k & \text{if } j = p+1. \end{cases}$$

Proof. The number of strings in $\mathbf{X}_{n,k}$ whose largest value in the first $i-1$ positions is p is $\left\{{i-1 \atop p}\right\}X_{n-i+2,k,p}$. The number of strings that, in addition, have a j in position i is $\left\{{i-1 \atop p}\right\}X_{n-i+1,k,p}$ if $0 \leq j \leq p$, and is $\left\{{i-1 \atop p}\right\}X_{n-i+1,k,p+1}$ if $j = p+1$. To complete the proof, divide each of the latter numbers by the first and apply Lemma 5.3 to the respective quotients. □

Let $\mathbf{s}'$ be an infinite length k-ary string chosen uniformly at random and $\mathbf{s} = can(\mathbf{s}')$. Note that

$$Prob(s_i = j \mid \max(s_1, \ldots, s_{i-1}) = p) \;=\; \begin{cases} 1/k & \text{if } 0 \le j \le p, \\ 1-(p+1)/k & \text{if } j = p+1. \end{cases}$$

exactly the same as the probabilities given in Lemma 5.4 above.

Let $\mathbf{s}'$ and $\mathbf{t}'$ be two infinite k-ary strings chosen uniformly at random and let $\mathbf{s} = can(\mathbf{s}') = s_1 s_2 s_3 \cdots$ and $\mathbf{t} = can(\mathbf{t}') = t_1 t_2 t_3 \cdots$. We have proven that

$$A_k = \lim_{n\to\infty} A_{n,k} \;=\; \sum_{i\ge 1} i \cdot Prob(c(\mathbf{s},\mathbf{t}) = i). \tag{5.4}$$

Computing and analyzing A_k using (5.4) is the subject of the next two subsections.

5.1. The value of A_k. The purpose of this section is to prove the following rather attractive expression for A_k (below, the notation $(k)_j^2$ means $[(k)_j]^2$, the square of the falling factorial).

THEOREM 5.5.

$$A_k \;=\; \sum_{j=1}^{k} \frac{(k)_j^2}{(k^2-1)_j}.$$

Select $\mathbf{s}'$ and $\mathbf{t}'$ uniformly at random from the set of infinite length k-ary strings and let $\mathbf{s} = can(\mathbf{s}')$ and $\mathbf{t} = can(\mathbf{t}')$. Denote by β_p the probability that the first p symbols of $\mathbf{s}$ and $\mathbf{t}$ agree.

$$\beta_i \;\stackrel{\text{def}}{=}\; Prob(s_1 = t_1, s_2 = t_2, \ldots, s_i = t_i). \tag{5.5}$$

By classifying the prefix of the first i symbols of $\mathbf{s}'$ and $\mathbf{t}'$ according to j, the number of distinct symbols in that prefix, we obtain the following expression for β_i.

$$\beta_i \;=\; \frac{1}{k^{2i}} \sum_{j=1}^{k} (k)_j^2 \left\{ {i \atop j} \right\}.$$

By (5.4),

$$\begin{aligned} A_k &= \sum_{i\ge 0} i \cdot Prob(s_1 = t_1, s_2 = t_2, \ldots, s_i = t_i, s_{i+1} \ne t_{i+1}) \\ &= \sum_{i\ge 0} i \cdot (\beta_i - \beta_{i+1}) \\ &= \sum_{i\ge 0} i \sum_{j=1}^{k} (k)_j^2 \left(\frac{\left\{ {i \atop j} \right\}}{k^{2i}} - \frac{\left\{ {i+1 \atop j} \right\}}{k^{2i+2}} \right) \qquad (5.6) \\ &= \sum_{j=1}^{k} (k)_j^2 \sum_{i\ge 0} \frac{1}{k^{2i}} \left\{ {i \atop j} \right\}. \end{aligned}$$

The inner sum of (5.7) can be simplified by using the following identity from Wilf [18, equation (1.6.5), page 19].

$$\sum_n \left\{ {n \atop k} \right\} x^n = \frac{x^k}{(1-x)(1-2x)\cdots(1-kx)}. \tag{5.7}$$

Setting x to k^{-2} and k to j in (5.7) and substituting into (5.7) finishes the proof of Theorem 5.5. It is remarkable that A_k has such a simple expression; perhaps there is a direct combinatorial proof.

5.2. The asymptotic value of A_k. Our goal in this section is to prove the following theorem.

THEOREM 5.6.

$$A_n = \frac{1}{2}\sqrt{\pi \cdot n} - \frac{1}{6} + o(1).$$

In this section we will use A_n instead of A_k as is customary in the asymptotics literature. Our main technique is called "trading tails," from Graham, Knuth, and Patashnik [4, starting at page 452]. In principle, our techniques could be applied to obtain additional terms in the asymptotic expansion of A_n. With this in mind, some of the expressions below are given with greater generality than is strictly necessary to prove Theorem 5.6.

We start by presenting Taylor's expansion of $\ln(1-z)$ and an identity involving the Bernoulli numbers. The expansion below is valid for $z \to 0$ (see page 438 of [4]). This is just Taylor's expansion with remainder.

$$\ln(1-z) = -\sum_{k=1}^{p} \frac{z^k}{k} + O(z^{p+1}). \tag{5.8}$$

From *Concrete Mathematics* [4, exercise 9.30, page 477, with solution on page 570],

$$\sum_{r\geq 0} r^{l-1} e^{-r^2/n} = \frac{1}{2} n^{l/2} \Gamma\left(\frac{l}{2}\right) - \sum_{k=0}^{p} \frac{(-1)^k B_{l+2k}}{n^k (l+2k)! k!} + O\left(\frac{1}{n^{p+1}}\right), \tag{5.9}$$

where $B_0, B_1, B_2, B_3, B_4, \ldots = 1, -\frac{1}{2}, \frac{1}{6}, 0, -\frac{1}{30}, 0, \ldots$ are the Bernoulli numbers (see, for instance, section 6.5 of [4]) and Γ is the gamma function with $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(3/2) = \frac{1}{2}\sqrt{\pi}$, $\Gamma(5/2) = \frac{3}{4}\sqrt{\pi}$, and $\Gamma(l) = (l-1)!$ if l is an integer.

In particular, setting $l = 1, 2, 4, 5$ in (5.9) yields

$$\sum_{r\geq 1} e^{-r^2/n} = \frac{1}{2}\sqrt{\pi n} - \frac{1}{2} + o(1), \tag{5.10}$$

$$\frac{1}{n} \sum_{r\geq 1} r e^{-r^2/n} = \frac{1}{2} + o(1), \tag{5.11}$$

$$\frac{1}{n^2} \sum_{r\geq 1} r^3 e^{-r^2/n} = \frac{1}{2} + o(1), \tag{5.12}$$

$$\frac{1}{n^3}\sum_{r\geq 1} r^4 e^{-r^2/n} = o(1). \tag{5.13}$$

Recall that

$$A_n = \sum_{r=1}^{n} \frac{(n)_r^2}{(n^2-1)_r} = \sum_{r=1}^{n}\prod_{x=0}^{r-1}\frac{(n-x)^2}{n^2-x-1},$$

and define

$$T(n,r) \stackrel{\text{def}}{=} \frac{n^2}{n^2-r}\prod_{x=1}^{r-1}\frac{(n-x)^2}{n^2-x} = \frac{1}{1-\frac{r}{n^2}}\prod_{x=1}^{r-1}\frac{(1-\frac{x}{n})^2}{1-\frac{x}{n^2}}. \tag{5.14}$$

For any small fixed $\epsilon > 0$ we may write A_n as

$$A_n = \sum_{r=1}^{n} T(n,r) = \sum_{r=1}^{n^{1/2+\epsilon}} T(n,r) \; + \sum_{r=1+n^{1/2+\epsilon}}^{n} T(n,r). \tag{5.15}$$

Take logarithms of $T(n,r)$ as expressed in the right-hand side of (5.14) and then use (5.8) to obtain the following equality, valid for any $r,\ r/n \to 0$, and any natural numbers p and q:

$$\begin{aligned}\ln(T(n,r)) &= 2\sum_{x=1}^{r-1}\ln\left(1-\frac{x}{n}\right) - \sum_{x=1}^{r}\ln\left(1-\frac{x}{n^2}\right)\\ &= \left(-2\sum_{k=1}^{p}\frac{1}{kn^k}\sum_{x=1}^{r-1}x^k + O\left(\frac{r^{p+2}}{n^{p+1}}\right)\right) + \left(\sum_{k=1}^{q}\frac{1}{kn^{2k}}\sum_{x=1}^{r}x^k + O\left(\frac{r^{q+2}}{n^{2q+2}}\right)\right).\end{aligned}$$

Setting $p=2$ and $q=0$ in the above expression, we obtain

$$\begin{aligned}\ln(T(n,r)) &= \left(-\frac{r(r-1)}{n} - \frac{r(r-1)(2r-1)}{6n^2} + O\left(\frac{r^4}{n^3}\right)\right) + \left(O\left(\frac{r^2}{n^2}\right)\right)\\ &= -\frac{r^2}{n} + \frac{r}{n} - \frac{r^3}{3n^2} + O\left(\frac{r^4}{n^3}\right).\end{aligned}$$

Thus, exponentiating the above and using the power series expansion of e^x,

$$\begin{aligned}T(n,r) &= \exp\left(-\frac{r^2}{n}\right)\exp\left(\frac{r}{n}\right)\exp\left(-\frac{r^3}{3n^2}\right)\exp\left(O\left(\frac{r^4}{n^3}\right)\right)\\ &= \exp\left(-\frac{r^2}{n}\right)\left(1+\frac{r}{n}+O\left(\frac{r^2}{n^2}\right)\right)\left(1-\frac{r^3}{3n^2}+O\left(\frac{r^6}{n^4}\right)\right)\left(1+O\left(\frac{r^4}{n^3}\right)\right)\\ &= \exp\left(-\frac{r^2}{n}\right)\left(1+\frac{r}{n}-\frac{r^3}{3n^2}+O\left(\frac{r^4}{n^3}\right)\right).\end{aligned}$$

Substituting the four expressions (5.10)–(5.13) into the sum $\sum_r T(n,r)$ and using the last derived expression for $T(n,r)$, we obtain

$$\begin{aligned}\sum_{r\geq 1} T(n,r) &= \sum_{r\geq 1}\exp\left(-\frac{r^2}{n}\right)\left(1+\frac{r}{n}-\frac{r^3}{3n^2}+O\left(\frac{r^4}{n^3}\right)\right)\\ &= \frac{1}{2}\sqrt{n\pi} - \frac{1}{6} + o(1).\end{aligned} \tag{5.16}$$

For any fixed integer j, small $\epsilon > 0$, $n^{1/2+\epsilon} \le r \le n$, and large enough n, note that

$$r^j e^{-r^2/n} < e^{-r^2/(2n)}.$$

Thus (following [4, page 474]),

$$\begin{aligned}\sum_{r>n^{1/2+\epsilon}} r^j e^{-r^2/n} &< \sum_{r>n^{1/2+\epsilon}} e^{-r^2/(2n)} \\ &< \exp(-\lfloor n^{1/2+\epsilon}\rfloor^2/(2n))(1 + e^{-1/(2n)} + e^{-2/(2n)} + \cdots) \\ &= O(e^{-\frac{1}{2}n^{2\epsilon}}) \cdot O(n) \\ &= O(n^{-M}) \qquad \text{for any fixed } M.\end{aligned}$$

Substituting $j = 0, 1, 3, 4$ into (5.16), we get

$$\sum_{r=1}^{n^{1/2+\epsilon}} T(n,r) = \frac{1}{2}\sqrt{n\pi} - \frac{1}{6} + o(1). \tag{5.17}$$

It remains only to show that the second sum on the right-hand side of (5.15) is $o(1)$.

LEMMA 5.7. *For all $j \ge 0$ and $\epsilon > 0$,*

$$\lim_{n\to\infty} n^j \left(1 - \frac{1}{n}\right)^{n^{1+\epsilon}} = 0.$$

Proof. Rewriting and exploiting continuity, we obtain

$$\begin{aligned}&\exp\left(\lim_{n\to\infty} \ln\left(n^j \left(1 - \frac{1}{n}\right)^{n^{1+\epsilon}}\right)\right) \\ &\quad = \exp\left(\lim_{n\to\infty} (j \ln n + n^{1+\epsilon}(\ln(n-1) - \ln n))\right) \\ &\quad = \exp\left(\lim_{n\to\infty} \frac{jn^{-(1+\epsilon)} \ln n + \ln(n-1) - \ln n}{n^{-(1+\epsilon)}}\right).\end{aligned}$$

The limit may now be evaluated by using L'Hopital's rule.

$$\begin{aligned}&= \exp\left(\lim_{n\to\infty} \frac{j(n^{-(2+\epsilon)} - (1+\epsilon)n^{-(2+\epsilon)} \ln n) + \frac{1}{n-1} - \frac{1}{n}}{-(1+\epsilon)n^{-(2+\epsilon)}}\right) \\ &= \exp\left(\lim_{n\to\infty} \left(-\frac{j}{1+\epsilon} + j \ln n - \frac{n^{2+\epsilon}}{n(n-1)(1+\epsilon)}\right)\right) \\ &= \exp\left(\lim_{n\to\infty} \left(O(1) + o(n^\epsilon) + \frac{n^\epsilon}{(n-1)(1+\epsilon)} - \frac{n^\epsilon}{1+\epsilon}\right)\right) \\ &= \exp\left(\lim_{n\to\infty} \left(o(n^\epsilon) - \frac{n^\epsilon}{1+\epsilon}\right)\right) = 0. \qquad \square\end{aligned}$$

LEMMA 5.8. *If* $r \geq n^{1/2+\epsilon}$ *where* $\epsilon > 0$, *then*

$$T(n,r) = O(n^{-M}), \quad \textit{for any fixed } M.$$

Proof. Note that the quantities $T(n,r)$ are monotonically decreasing in r, and thus we need only consider $T(n, n^{1/2+\epsilon})$. First, split the definitional product (5.14) into two factors and then bound each factor.

$$T(n,n^{1/2+\epsilon}) = \frac{n^2}{n^2-n^{1/2+\epsilon}} \prod_{x=1}^{\lfloor \frac{1}{2}n^{1/2+\epsilon} \rfloor} \frac{(n-x)^2}{n^2-x} \prod_{x=\lfloor \frac{1}{2}n^{1/2+\epsilon} \rfloor+1}^{n^{1/2+\epsilon}} \frac{(n-x)^2}{n^2-x}.$$

$$\begin{aligned} T(n,n^{1/2+\epsilon}) &\leq 2\left(\frac{(n-\frac{1}{2}n^{1/2+\epsilon})^2}{n^2-\frac{1}{2}n^{1/2+\epsilon}}\right)^{\frac{1}{2}n^{1/2+\epsilon}} \\ &= 2\left(1-\frac{n^{3/2+\epsilon}-\frac{1}{4}n^{1+2\epsilon}-\frac{1}{2}n^{1/2+\epsilon}}{n^2-\frac{1}{2}n^{1/2+\epsilon}}\right)^{\frac{1}{2}n^{1/2+\epsilon}}. \end{aligned}$$

This last expression grows like this,

$$2\left(1-\frac{1}{n^{1/2-\epsilon}}\right)^{\frac{1}{2}n^{1/2+\epsilon}},$$

which by Lemma 5.7 is $O(n^{-M})$ for any fixed M. □

From Lemma 5.8 it follows that

$$\sum_{r=1+n^{1/2+\epsilon}}^{n} T(n,r) \;=\; O(n^{-M}), \quad \text{for any fixed } M.$$

This last equation, together with (5.17), proves Theorem 5.6.

6. Application: generating *k*-paths. A generalization of trees yields the notion of *k-trees* as the skeletons of acyclic simplicial complexes for dimensions higher than 2. For a fixed k, the complete graph with $k+1$ vertices, K_{k+1}, is a k-tree, and every k-tree with more vertices can be constructed from a smaller k-tree by adding a new node and making it adjacent to all vertices of a K_k subgraph. Thus, every k-tree has at least two vertices of degree k, *k-leaves*, and every *minimal separator* (a set of vertices disconnecting the graph, minimal with respect to set inclusion) consists of k mutually adjacent vertices [13]. (We will say that this set *induces* K_k.) In the simplest case of a k-tree with exactly two k-leaves, we have a generalization of a path, namely a *k-path*. The two k-leaves of a k-path (say, u and v) are connected by k unique vertex-disjoint paths and every minimal separator contains exactly one vertex of each path [11]. One can view the construction process of such a graph as starting with one k-leaf, say u, of K_{k+1}, and then adding vertices until the other desired k-leaf, say v, is added, completing the construction. Recording which of the (eventual) vertex disjoint paths are augmented during the process gives the unique string over an alphabet of k symbols corresponding to the particular construction process (see Figure 6.1). A given k-path can be constructed in this manner from either of its ends (from u or from v) and vertex-disjoint paths can be identified by any of the $k!$ permutations of symbols $0, 1, \ldots, k-1$. Thus the problem of listing all nonisomorphic k-paths can be dealt with as the problem of listing all equivalence classes of strings of k symbols, under the group actions of string reversal and/or symbol permutation.

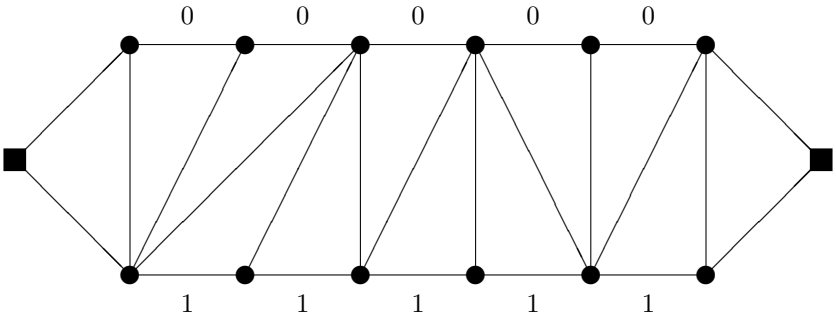


FIG. 6.1. *The* 2*-path corresponding to the binary string* 0011011001.

7. Final remarks. We list below some open problems that are inspired by or related to the problems considered in this paper.

1. Is there a direct combinatorial proof of the simple expression in Theorem 1 for A_k?

2. Is there a CAT BEST algorithm for generating the elements of $\mathbf{Y}_{n,k}$ when $k > 2$? The results of section 2 show that $\mathbf{Y}$ is a context-free language for $k = 2$ and it is an unambiguous grammar for this language that leads us to the CAT algorithm. Can such a grammar be found for larger values of k?

3. Can these results be extended to generate all non-isomorphic k-trees? Here, even the counting problem is unsolved. Related enumeration (counting) results are reported in Harary and Palmer [5] and Hering, Read, and Shephard [6].

4. Does there exist a CAT algorithm to generate all k-ary strings where the group acting on the string contains rotations as well as permutations of the symbols and/or reversal of the string? The lexicographically smallest strings in classes of strings equivalent under rotations are referred to as necklaces. These are useful, for example, in generating de Bruijn sequences.

Acknowledgments. We wish to thank Andrew Odzlyko, and also Philippe Flajolet and Bruno Salvy, for their help in proving Theorem 5.6. They responded to a query posted to the internet newsgroup (sci.math.research). We also wish to thank Peter Winkler for helpful discussions and the referees for carefully reading the paper and making many useful suggestions.

REFERENCES

[1] M. C. ER, *A fast algorithm for generating set partitions*, Comput. J., 31 (1988), pp. 283–284.
[2] H. FREDRICKSEN AND I. J. KESSLER, *An algorithm for generating necklaces of beads in two colors*, Discrete Math., 61 (1986), pp. 181–188.
[3] H. FREDRICKSEN AND I. J. MAIORANA, *Necklaces of beads in k colors and k-ary de Bruijn sequences*, Discrete Math., 23 (1978), pp. 207–210.
[4] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics*, Addison-Wesley, Reading, MA, 1989.
[5] F. HARARY AND E. M. PALMER, *On acyclic simplicial complexes*, Mathematika, 15 (1968), pp. 115–122.
[6] F. HERING, R. C. READ, AND G. C. SHEPHARD, *The enumeration of stack polytopes and simplicial clusters*, Discrete Math., 40 (1982), pp. 203–217.
[7] G. HUTCHINSON, *Partitioning algorithms for finite sets*, Comm. ACM, 6 (1963), pp. 613–614.
[8] R. A. KAYE, *A Gray code for set partitions*, Inform. Process. Lett., 5 (1976), pp. 171–173.

[9] D. E. Knuth, *The Art of Computer Programming, Vol.* I: *Fundamental Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1973.
[10] H. R. Lewis and L. Denenberg, *Data Structures & Their Algorithms*, Harper-Collins, New York, 1991.
[11] A. Proskurowski, *Separating subgraphs in k-trees: Cables and caterpillars*, Discrete Math., 49 (1984), pp. 275–285.
[12] A. Proskurowski, F. Ruskey, and M. Smith, *Analysis of algorithms for generating k-paths*, Tech. Report DCS-178, Univ. of Victoria, British Columbia, 1992.
[13] D. J. Rose, *Triangulated graphs and the elimination process*, J. Math. Anal. Appl., 32 (1970), pp. 597–609.
[14] F. Ruskey, C. D. Savage, and T. Wang, *Generating necklaces*, J. Algorithms, 13 (1992), pp. 414–430.
[15] F. Ruskey, *Listing all bitstrings inequivalent under reversal*, manuscript, 1990.
[16] I. Semba, *An efficient algorithm for generating all partitions of the set* $\{1, 2, \ldots, n\}$, J. Inform. Process., 7 (1984), pp. 41–42.
[17] D. Stanton and D. White, *Constructive Combinatorics*, Springer-Verlag, Berlin, 1986.
[18] H. S. Wilf, *Generating Functionology*, Academic Press, New York, 1990.
[19] S. G. Williamson, *Combinatorics for Computer Science*, Computer Science Press, Rockville, MD, 1985.
[20] S. G. Williamson, *Ranking algorithms for lists of partitions*, SIAM J. Comput., 5 (1976), pp. 602–617.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 110–127, February 1998

008

AN ORDERING ON THE EVEN DISCRETE TORUS*

OLIVER RIORDAN†

Abstract. The even discrete torus $T^n(k_1,\dots,k_n)$ is the graph on $(\mathbb{Z}/k_1\mathbb{Z})\times\cdots\times(\mathbb{Z}/k_n\mathbb{Z})$, where each k_i is even and $\mathbf{x}=(x_1,\dots,x_n)$ is joined to $\mathbf{y}=(y_1,\dots,y_n)$ if for some i we have $x_i=y_i\pm1$ and $x_j=y_j$ for all $j\neq i$. The main aim of this paper is to describe an ordering on the even discrete torus whose initial segments give a best possible isoperimetric inequality. This extends the partial solution of Bollobás and Leader [*SIAM J. Discrete Math.*, 3 (1990), pp. 32–37] to a problem posed by Wang and Wang [*SIAM J. Appl. Math.*, 33 (1977), pp. 55–59].

Key words. isoperimetric inequality, discrete torus

AMS subject classification. 05C35

PII. S0895480194278234

1. Introduction.

For $n\geq 1$ and $2\leq k_1\leq\cdots\leq k_n$, the *discrete torus* $T^n=T^n(k_1,\dots,k_n)$ is the graph on the set $\mathbb{Z}_{k_1}\times\cdots\times\mathbb{Z}_{k_n}$ in which $\mathbf{x}=(x_1,\dots,x_n)$ and $\mathbf{y}$ are adjacent if there is some i with $x_i=y_i\pm1$ and $x_j=y_j$ for $j\neq i$. Alternatively, T^n is the product of n cycles of lengths $k_1,\dots,k_n$.

Given a set X of vertices of a graph G, define the *neighborhood* of X as

$$N(X)=\{\mathbf{x}\in V(G):\mathbf{x}\in X \text{ or } \exists\mathbf{y}\in X \text{ with } \mathbf{xy}\in E(G)\}.$$

Define the *boundary* of X to be $\partial X=N(X)\backslash X$.

An inequality of the form

$$|N(X)|\geq g(a) \text{ whenever } X\subseteq V(G) \text{ and } |X|=a$$

is called an *isoperimetric inequality* on G. For a graph G one would like to determine the best possible isoperimetric inequality for every a, i.e., the function

$$f(a)=\min\{|N(X)|:X\subseteq V(G),|X|=a\},$$

and ideally to describe the *extremal* sets, i.e., the sets $X\subseteq V(G)$ such that $|X|=a$ and $|N(X)|=f(a)$ for given a.

An example of such an inequality is Harper's theorem [3], where the graph is the *discrete cube*, $\{0,1\}^n$. This has been generalized to $\mathbb{Z}^n$ by Wang and Wang [4], and to the product of n paths of length h by Bollobás and Leader [2]. Wang and Wang also posed exactly this problem in the discrete torus [5], and for the case $k_1=\cdots=k_n=2h$ it has been answered by Bollobás and Leader [1] for some values of a. The aim of this paper is to extend this by proving the following result.

THEOREM 1.1. *Let $n\geq 1$ and $4\leq k_1\leq\cdots\leq k_n$, where each k_i is even. Then there is an ordering $\prec$ on $T^n=T^n(k_1,\dots,k_n)$ with the property that whenever $X\subseteq T^n$ and C is the initial segment of $(T^n,\prec)$ with $|C|=|X|$, we have $|N(X)|\geq|N(C)|$.*

*Received by the editors December 5, 1994; accepted for publication (in revised form) January 6, 1997.

http://www.siam.org/journals/sidma/11-1/27823.html

†Department of Pure Mathematics and Mathematical Statistics, 16 Mill Lane, Cambridge CB2 1SB, UK (omr10@cus.cam.ac.uk).

Such an ordering need not exist in general—the odd torus $\mathbb{Z}_3^3$ provides a counterexample, as there are no nested extremal sets of sizes 7 and 8.

In fact Theorem 1.1 will be proved for a specific ordering, defined in section 4, whose initial segments include the sets $\{\mathbf{x} \in T^n : d(\mathbf{x}, 0) \leq r\}$, so we can deduce that if $X \subseteq T^n$ satisfies

$$|X| = |\{\mathbf{x} \in T^n : d(\mathbf{x}, 0) \leq r\}|,$$

then

$$|N(X)| \geq |\{\mathbf{x} \in T^n : d(\mathbf{x}, 0) \leq r + 1\}|.$$

Since the neighborhood of an initial segment is again an initial segment (see section 5), we can apply Theorem 1.1 several times in succession. In particular, setting $N^0(X) = X$, $N^{m+1}(X) = N(N^m(X))$ $(m \geq 0)$, so

$$N^m(X) = \{\mathbf{x} \in V(G) : \exists \mathbf{y} \in X \text{ with } d(\mathbf{x}, \mathbf{y}) \leq m\},$$

we have the following corollary.

COROLLARY 1.2. *Under the conditions of Theorem* 1.1, $|N^m(X)| \geq |N^m(C)|$.

We shall also prove (in section 8) that the complement of an initial segment is isomorphic to an initial segment. Since the neighborhood of the complement of a set X is the complement of the interior of X, $int\, X$, we have $N^m(X^c) = (int^m(X))^c$, and from Corollary 1.2 we can deduce a corresponding result for interiors.

COROLLARY 1.3. *Under the conditions of Theorem* 1.1, $|int^m(X)| \leq |int^m(C)|$.

The ordering used will be a natural adaptation of that given by Wang and Wang [4] for $\mathbb{Z}^n$: T^n is considered as a particular subset of $\mathbb{Z}^n$, and their order is applied to this. It is rather surprising that this works, i.e., that when the wraparound in the torus is taken into account it is not necessary to move a single point to minimize the new neighborhood.

2. Outline of proof. Most of the paper is devoted to the proof of Theorem 1.1, so before we get down to work we describe how the proof is organized.

At various stages we shall consider a close relative of the torus, the *grid*—the graph $\{0, 1, \ldots, h-1\}^n$ (i.e., the product of n paths of length h, or more generally of lengths $h_1, \ldots, h_n$). In [2], Bollobás and Leader prove an isoperimetric inequality on the grid, and although this will not be used, a related result concerning the isoperimetric function will be. This is of interest in its own right, and constitutes section 3.

In section 4, we define a total ordering $\prec$ on the even torus T^n and state the main result—that initial segments of this ordering have the smallest possible neighborhoods given their size. In section 5, we start to examine the properties of this ordering, showing that the neighborhood of an initial segment is always an initial segment. To do this we use a decomposition of T^n into 2^n copies of the grid (with side lengths half those of the torus) and the fact that on each of these copies $\prec$ reduces to the *simplicial* order of [2], which has a corresponding property.

The main idea of the proof is *compression* (section 6), i.e., tidying up a set X without increasing its neighborhood, until it becomes sufficiently tidy to compare with the initial segment of the same size. For this we use a different decomposition of the torus. For each i, T^n can be considered as the union of k_i copies of an $n-1$-dimensional torus, so we can use induction: within each copy we replace the points of X by the initial segment of T^{n-1} of the same size. Using the result of section 5, we show that this does not increase the neighborhood.

Because for every set X there is a compressed set with no larger neighborhood, we only need to compare the neighborhoods of compressed sets and initial segments (which are also compressed) of the same size. We start by looking at the neighborhood of a compressed set (section 7).

Unfortunately, for $n = 2$ the fact that a set $X \subseteq T^2$ is compressed does not give that much information about the structure of X. This case of the proof is separate from the rest, and consists mainly of tedious case-checking. This is given in a compressed form as an appendix. For $n \geq 3$ it turns out that we only need to consider the *n-slices* (i.e., planes $x_n =$ constant) $0, 1$, and $-h+1, h$ of the compressed set and the corresponding initial segment. The amount of work is further reduced by using a symmetry ρ of the torus, introduced in section 8. This is a key stage in the proof—at this point we will have shown that it is sufficient to prove that for $X \subseteq T^n$ compressed and C the initial segment with $|C| = |X|$,

$$|N(C_0)| + |N(C_1)| \leq |N(X_0)| + |N(X_1)|, \tag{2.1}$$

and

$$|C_0| + |C_1| \leq |X_0| + |X_1|, \tag{2.2}$$

where X_t is the slice $x_n = t$ of X, considered as a subset of T^{n-1}.

The next step is to examine the possibilities for these slices—they turn out to be close to those of an initial segment, in that moving a few points (in a sense made precise in section 9) from one to the other gives the middle slices of an initial segment. Projecting onto the grid, we show in section 10 that this can be done consistently with (2.1), using Lemma 3.1 from section 3.

At this point, we have something like (2.1) and (2.2), but with C_0, C_1 replaced by D_0, D_1, the middle two slices of some initial segment which need not be the same size as X. The final step is to give a method of constructing a set from two slices which gives an initial segment where possible, and to show that the set D constructed from D_0, D_1 is at least as large as X and hence C. This implies that $C \subseteq D$ and hence $C_t \subseteq D_t$ $(t = 0, 1)$, so we can deduce (2.1) and (2.2), completing the proof.

3. A result in the grid. Let $n \geq 1$ and $1 \leq h_1 \leq \cdots \leq h_n$, and let $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$ be two points of the grid $[h_1] \times \cdots \times [h_n]$. Let $|\mathbf{x}| = x_1 + \cdots + x_n$. Then $\mathbf{x} \prec \mathbf{y}$ in the *simplicial* order means

$$\begin{aligned} &|\mathbf{x}| < |\mathbf{y}|, \\ \text{or } &|\mathbf{x}| = |\mathbf{y}| \text{ and } x_i > y_i \text{ for } i = \min\{j : x_j \neq y_j\}. \end{aligned}$$

This ordering can be described in words by: work upwards in *layers* (sets with $|\mathbf{x}|$ constant), and within each layer first maximize x_1, for fixed x_1 first maximize x_2, and so on.

In [2], Bollobás and Leader showed that initial segments of the simplicial order minimize the neighborhood among sets of a given size, but this will not be used here. Instead we shall give a result about how the sizes of the neighborhoods of these initial segments relate to each other, which will be needed later.

Let n, $h \geq 1$ and let $\mathbf{x}_1, \ldots, \mathbf{x}_{h^n}$ be the points of the grid $[h]^n$ in simplicial order. Let $C(m) = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be the initial segment of size m. We shall say that $\mathbf{x}_m$ adds the point $\mathbf{y}$ to the neighborhood if

$$\mathbf{y} \in N(C(m)) \backslash N(C(m-1)),$$

i.e., if $\mathbf{x}_m$ is the earliest neighbor of $\mathbf{y}$ in the simplicial order. Let $a_n(m)$ be the number of points added to the neighborhood by $\mathbf{x}_m$.

LEMMA 3.1. *Suppose that* $0 \le i \le j \le h^n - l$ *and, if* $l > 0$, *either*

(i) $\mathbf{x}_{i+1}$ *is the first point in a layer; or*

(ii) $\mathbf{x}_{j+l}$ *is the last point in a layer.*

Then

$$\sum_{k=1}^{l} a_n(i+k) \ge \sum_{k=1}^{l} a_n(j+k).$$

Note that although we have taken $h_1 = \cdots = h_n$ for simplicity, the result remains true and the proof is essentially unmodified for $1 \le h_1 \le \cdots \le h_n$.

It turns out to be easier to work with the following definition.

Let $A'_n(\mathbf{x}_m)$ be the set of points in layer $|\mathbf{x}_m| + 1$ added to the neighborhood by $\mathbf{x}_m$, and let $a'_n(m) = |A'_n(\mathbf{x}_m)|$.

Only $\mathbf{0}$ adds a point in its own layer and, being the first point, it can only appear among $\mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+l}$ if $j = 0$, whence $i = 0$ also, so Lemma 3.1 follows from Lemma 3.2 below.

LEMMA 3.2. *Under the conditions of Lemma* 3.1,

$$\sum_{k=1}^{l} a'_n(i+k) \ge \sum_{k=1}^{l} a'_n(j+k). \tag{3.1}$$

Note that this new definition is as natural as the previous one—when considering points in one layer, their neighbors in the next are a generalization of the upper shadow of the Kruskal-Katona theorem. The reason for stating Lemma 3.1 is that it is this form which will be useful, and the reason for proving Lemma 3.2 is that it is slightly stronger, and this is needed for the induction to work.

Proof of Lemma 3.2. We use induction on n, and for fixed n induction on l. The case $l = 0$ is trivial. If $n = 1$ then $a'_n(m) = \begin{cases} 1 & m < h \\ 0 & m = h \end{cases}$ which is decreasing, so $i \le j$ implies $a'_n(i+k) \ge a'_n(j+k)$ and hence (3.1).

Now if $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+l}\}$ and $\{\mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+l}\}$ overlap, i.e., if $j = i + r$ for some $r < l$, then to prove (3.1) we only need to compare the sums of a'_n over the two smaller sets $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+r}\}$ and $\{\mathbf{x}_{j+l-r+1}, \ldots, \mathbf{x}_{j+l}\}$. We can do this by induction on l, as whichever of (i) or (ii) holds is also satisfied by the smaller sets. We may thus assume that $i + l < j + 1$.

If either $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+l}\}$ or $\{\mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+l}\}$ contains points from more than one layer, we are again done by induction. This time there are four cases.

Suppose (i) holds and there is a layer break between $\mathbf{x}_{i+r}$ and $\mathbf{x}_{i+r+1}$. Then we can apply the induction hypothesis to the two pairs of sets $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+r}\}$, $\{\mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+r}\}$ and $\{\mathbf{x}_{i+r+1}, \ldots, \mathbf{x}_{i+l}\}$, $\{\mathbf{x}_{j+r+1}, \ldots, \mathbf{x}_{j+l}\}$, as we have that $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+r+1}$ are each the first point in some layer.

If (i) holds and there is a layer break between $\mathbf{x}_{j+r}$ and $\mathbf{x}_{j+r+1}$, we can use the two pairs of sets $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+l-r}\}$, $\{\mathbf{x}_{j+r+1}, \ldots, \mathbf{x}_{j+l}\}$ and $\{\mathbf{x}_{i+l-r+1}, \ldots, \mathbf{x}_{i+l}\}$, $\{\mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+r}\}$. The first pair satisfies the conditions for the induction hypothesis as $\mathbf{x}_{i+1}$ is the first point in a layer, and the second pair as $\mathbf{x}_{j+r}$ is the last point in a layer.

If (ii) holds we have the mirror image of the above two cases. If we have a layer break in $\{\mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+l}\}$, say between $\mathbf{x}_{i+l-r}$ and $\mathbf{x}_{i+l-r+1}$, we again use the pairs

$\{\mathbf{x}_{i+1},\dots,\mathbf{x}_{i+l-r}\}$, $\{\mathbf{x}_{j+r+1},\dots,\mathbf{x}_{j+l}\}$ and $\{\mathbf{x}_{i+l-r+1},\dots,\mathbf{x}_{i+l}\}$, $\{\mathbf{x}_{j+1},\dots,\mathbf{x}_{j+r}\}$. This time, the induction hypothesis applies as $\mathbf{x}_{j+l}$ is the last point in a layer, and $\mathbf{x}_{i+l-r+1}$ is the first.

Finally, if (ii) holds and there is a layer break between $\mathbf{x}_{j+r}$ and $\mathbf{x}_{j+r+1}$, we use the pairs $\{\mathbf{x}_{i+1},\dots,\mathbf{x}_{i+r}\}$, $\{\mathbf{x}_{j+1},\dots,\mathbf{x}_{j+r}\}$ and $\{\mathbf{x}_{i+r+1},\dots,\mathbf{x}_{i+l}\}$, $\{\mathbf{x}_{j+r+1},\dots,\mathbf{x}_{j+l}\}$, as $\mathbf{x}_{j+r}$ and $\mathbf{x}_{j+l}$ are each the last point in a layer.

This leaves $\{\mathbf{x}_{i+1},\dots,\mathbf{x}_{i+l}\}$ contained in one layer, L_{r_-}, say, and $\{\mathbf{x}_{j+1},\dots,\mathbf{x}_{j+l}\}$ in another, L_{r_+}, with $r_+ \geq r_-$. For this case, we shall use induction on n.

In reducing the dimension by one the following notation will be useful. For $\mathbf{y} = (y_1,\dots,y_n)$ let $\overline{\mathbf{y}} = (y_2,\dots,y_n) \in [h]^{n-1}$. Let $\mathbf{e_i}$ be the vector with a one in the ith coordinate and zeros elsewhere. Now

$$\begin{aligned}\mathbf{y}+\mathbf{e_1} \in A'_n(\mathbf{y}) &\Leftrightarrow \begin{cases} y_1 < h-1, \\ y_i = 0, & i>1 \text{ (else } \mathbf{y}+\mathbf{e_1}-\mathbf{e_i} \prec \mathbf{y}) \end{cases} \\ &\Leftrightarrow \mathbf{y} \text{ is the first point in } L_{y_1} \text{ and } y_1 < h-1.\end{aligned}$$

Also $\mathbf{z} \in A'_n(\mathbf{y})$ and $\mathbf{z} \neq \mathbf{y}+\mathbf{e_1}$ imply $\mathbf{z} = \mathbf{y}+\mathbf{e_i}$ for some $i>1$, and if $\mathbf{z} = \mathbf{y}+\mathbf{e_i}$ for some $i>1$, then

$$\begin{aligned}\mathbf{z} \in A'_n(\mathbf{y}) &\Leftrightarrow \mathbf{z}\text{'s first neighbor is } \mathbf{y} \\ &\Leftrightarrow \mathbf{z}\text{'s first neighbor in the slice } \{\mathbf{w} : w_1 = y_1\} \text{ is } \mathbf{y} \\ &\qquad (\text{as } \mathbf{y} = \mathbf{z}-\mathbf{e_i} \prec \mathbf{z}-\mathbf{e_1}) \\ &\Leftrightarrow \bar{\mathbf{z}}\text{'s first neighbor is } \bar{\mathbf{y}} \\ &\Leftrightarrow \bar{\mathbf{z}} \in A'_{n-1}(\bar{\mathbf{y}}).\end{aligned}$$

Thus the points of $A'_n(\mathbf{y})$ can be divided up as follows:

(a) $\mathbf{y}+\mathbf{e_1} \in A'_n(\mathbf{y})$ iff $\mathbf{y}$ is the first point in some layer $r < h-1$,

(b) $\{\mathbf{z} : z_1 = y_1, \text{ and } \overline{\mathbf{z}} \in A'_{n-1}(\overline{\mathbf{y}})\}$.

We distinguish four cases, according to whether $r_+ = r_-$ or $r_+ > r_-$, and whether (i) or (ii) holds.

(A) $r_+ = r_-$, and $\mathbf{x}_{i+1}$ is the first point in this layer,

(B) $r_+ = r_-$, and $\mathbf{x}_{j+l}$ is the last point in this layer,

(C) $r_+ > r_-$, and $\mathbf{x}_{i+1}$ is the first point in layer L_{r_-},

(D) $r_+ > r_-$, and $\mathbf{x}_{j+l}$ is the last point in layer L_{r_+}.

Consider neighbors of type (a) first: these can only contribute more to the right-hand side of (3.1) than to the left if $\mathbf{x}_{j+1}$ starts layer L_{r_+}, $r_+ < h-1$ and $\mathbf{x}_{i+1}$ does not start L_{r_-}, which could only happen in case D. For it to happen in this case requires $\mathbf{x}_{j+1},\dots,\mathbf{x}_{j+l}$ to comprise all of L_{r_+}, implying $l = |L_{r_+}|$. But $\{\mathbf{x}_{i+1},\dots,\mathbf{x}_{i+l}\} \subseteq L_{r_-}$, so $l \leq |L_{r_-}|$ and $|L_{r_+}| \leq |L_{r_-}|$. Since until the "top boundaries" $y_i < h$ of the grid become relevant successive layers increase in size, this implies that $r_+ \geq h$, contradicting $r_+ < h-1$. (It is here that we use $h_1 \leq \cdots \leq h_n$, as we need $r_+ \geq \min\{h_i\}$ to imply $r_+ \geq h_1$.)

For neighbors of type (b), we have a bijection

$$\beta_r : \{\mathbf{x} \in [h]^n : |\mathbf{x}| = r\} \to \{\mathbf{x} \in [h]^{n-1} : r-(h-1) \leq |\mathbf{x}| \leq r\}$$

given by $\mathbf{x} \mapsto \bar{\mathbf{x}}$, which is order preserving. Thus β_{r_-} maps $\mathbf{x}_{i+1},\dots,\mathbf{x}_{i+l}$ to a block of l consecutive points in $[h]^{n-1}$, and similarly β_{r_+} $\mathbf{x}_{j+1},\dots,\mathbf{x}_{j+l}$. Also if $\mathbf{x}_{i+1}$ is the first point in a layer, then so is $\beta_{r_-}(\mathbf{x}_{i+1})$ (the first point in $im\beta_{r_-}$, a union of layers), and if $\mathbf{x}_{j+l}$ is the last point in a layer, so is $\beta_{r_+}(\mathbf{x}_{j+l})$.

Now if $r_+ = r_-$, then

$$\mathbf{x}_{i+1} \preceq \mathbf{x}_{j+1} \Rightarrow \beta_{r_-}(\mathbf{x}_{i+1}) \preceq \beta_{r_+}(\mathbf{x}_{j+1}).$$

If $r_+ \neq r_-$, then in case C, $\mathbf{x}_{i+1}$ is the first point of $im\beta_{r_-}$ and $\mathbf{x}_{j+1}$ is some point of $im\beta_{r_+}$, which starts at a later layer, so $\beta_{r_-}(\mathbf{x}_{i+1}) \preceq \beta_{r_+}(\mathbf{x}_{j+1})$, and in case D,

$$\begin{aligned} &\beta_{r_-}(\mathbf{x}_{i+l}) \preceq \beta_{r_+}(\mathbf{x}_{j+l}), \text{ similarly,} \\ \Rightarrow\ &\beta_{r_-}(\mathbf{x}_{i+1}) \preceq \beta_{r_+}(\mathbf{x}_{j+1}). \end{aligned}$$

(Note that some layers, e.g., L_{-1}, are empty, but this does not matter.)

Finally, by induction on n,

$$\sum_{k=1}^{l} |A'_{n-1}(\beta_{r_-}(\mathbf{x}_{i+k}))| \geq \sum_{k=1}^{l} |A'_{n-1}(\beta_{r_+}(\mathbf{x}_{j+k}))|,$$

and as the points of type (b) are in bijection with $A'_{n-1}(\beta_r(\mathbf{y}))$, this with the result for points of type (a) above gives (3.1), completing the proof of the induction step, and hence that of Lemma 3.1. □

4. The torus. Regarding $m \in \mathbb{Z}_k$, $k = 2h$, as an integer in the range $-h+1 \leq m \leq h$, let $|m|$ be the usual absolute value of m, and let $\sigma(m) = 1$ if $m > 0$, -1 if $m \leq 0$. For a vector $\mathbf{x} \in T^n$, let $|\mathbf{x}| = |x_1| + \cdots + |x_n|$. To define the ordering we introduce the following terminology.

The sth *layer* is $L_s = \{\mathbf{x} \in T^n : |\mathbf{x}| = s\}$,

the sth *Hamming ball* is $H_s = \{\mathbf{x} \in T^n : |\mathbf{x}| \leq s\}$,

and for $\beta \in \{+1, -1\}^n$, the *orthant* O_β is $\{\mathbf{x} \in T^n : \sigma(x_i) = \beta(i) \text{ for } i = 1, \ldots, n\}$. We define an ordering $\prec$ on the orthants by

$$O_\beta \prec O_\gamma \Leftrightarrow \beta(i) = +1, \gamma(i) = -1 \text{ for } i = \max\{j : \beta(j) \neq \gamma(j)\},$$

and on the discrete torus itself as follows.

Given $\mathbf{x} \in O_\alpha$ and $\mathbf{y} \in O_\beta$, $\mathbf{x} \prec \mathbf{y}$ if and only if

$$\begin{aligned} &|\mathbf{x}| < |\mathbf{y}|, \\ \text{or } &|\mathbf{x}| = |\mathbf{y}| \text{ and } O_\alpha \prec O_\beta, \\ \text{or } &|\mathbf{x}| = |\mathbf{y}|, O_\alpha = O_\beta, \text{ and } |x_i| > |y_i| \text{ for } i = \min\{j : x_j \neq y_j\}. \end{aligned}$$

In other words, within a layer, go through the faces in binary order. For each face, go through the points in simplicial order and include points on edges, corners, etc., iff all the other faces they meet have been included already. Thus an initial segment is just a set X for which there exist r and α such that

(i) $H_{r-1} \subseteq X \subseteq H_r$,

(ii) in layer r, X is full in orthants O_β preceding O_α (i.e., $L_r \cap O_\beta \subseteq X$) and empty in orthants O_β following O_α (i.e., $L_r \cap O_\beta \subseteq X^c$).

(iii) $X \cap O_\alpha$ is an initial segment of O_α.

For $X \neq \emptyset$, choosing L_r to be the last layer which meets X, and O_α the last orthant which meets $X \cap L_r$, we may also assume that

(iv) $X \cap O_\alpha \cap L_r \neq \emptyset$.

We can now state more precisely the aim of this paper, which is to prove that this ordering gives a best possible isoperimetric inequality.

THEOREM 4.1. *Let $n \geq 1$ and $4 \leq k_1 \leq \cdots \leq k_n$, where each k_i is even. Let $X \subseteq T^n = T^n(k_1, \ldots, k_n)$, and let C be the initial segment of $(T^n, \prec)$ with $|C| = |X|$. Then $|N(X)| \geq |N(C)|$.*

Before proving Theorem 4.1, we prove various results about the ordering $\prec$ which will be needed. Once we have done this, the proof itself (given in section 12) will be reasonably short. As most of the proof is almost unchanged when $k_1 = \cdots = k_n = k$, say, this will be assumed for most of what follows (to simplify the notation). When a substantial modification is needed to the proof to regain the generality stated above (as in the case $n = 2$), this will be indicated.

5. Neighborhoods of initial segments. Before introducing compression, we need to show that the neighborhood of an initial segment is another initial segment. To do this we use the decomposition of the torus into orthants isomorphic to the grid, and the corresponding result for the grid. Unfortunately there is some messy detail concerning the first point in each orthant.

We start by looking at just one orthant.

LEMMA 5.1. *Let C be an initial segment of $(\mathbb{Z}_k^n, \prec)$, $n \geq 1$, $k = 2h$, and O_α an orthant. Then*

$$N(C) \cap O_\alpha = N_{O_\alpha}(C \cap O_\alpha)$$

unless $C \cap O_\alpha = \emptyset$, when possibly

$$N(C) \cap O_\alpha = \{\text{1st point of } O_\alpha\}.$$

Here $N_{O_\alpha}(C \cap O_\alpha)$ is the neighborhood in the graph O_α of $C \cap O_\alpha$.

Proof. Clearly $N_{O_\alpha}(C \cap O_\alpha) \subseteq N(C) \cap O_\alpha$. Suppose $\mathbf{x} \in (N(C) \cap O_\alpha) \backslash N_{O_\alpha}(C \cap O_\alpha)$, i.e., $\mathbf{x}$ has a neighbor(s) in $C \cap O_\alpha^c$, but none in $C \cap O_\alpha$—and in particular $\mathbf{x} \in O_\alpha$, $\mathbf{x} \notin C$. Let $\mathbf{y} \in C$ be a neighbor of $\mathbf{x}$ in a different orthant. Then

$$\exists i \text{ s.t. } \left\{ \begin{array}{lll} (a) & x_i = 0 & y_i = 1, \\ \text{or } (b) & x_i = -h+1 & y_i = h, \\ \text{or } (c) & x_i = h & y_i = -h+1, \\ \text{or } (d) & x_i = 1 & y_i = 0, \end{array} \right\} \text{ and } x_j = y_j \text{ for } j \neq i.$$

If (a) or (b) holds, then $|\mathbf{y}| > |\mathbf{x}|$, so $\mathbf{x} \prec \mathbf{y}$ and as $\mathbf{y} \in C$, we have $\mathbf{x} \in C$—a contradiction.

If (c) holds let $\mathbf{z} = \mathbf{x} - \mathbf{e_i}$. Then $|\mathbf{z}| = |\mathbf{y}|$, but $\mathbf{z}$ is in an earlier orthant than $\mathbf{y}$ (as $\sigma(z_i) = +1$, $\sigma(y_i) = -1$), so $\mathbf{z} \prec \mathbf{y}$, and $\mathbf{z} \in C$. Thus $\mathbf{x}$ has a neighbor ($\mathbf{z}$) in $C \cap O_\alpha$—a contradiction.

If (d) holds and if $\mathbf{x}$ is not the first point of O_α, then for some j, $\mathbf{z} = \mathbf{x} - \sigma(x_j)\mathbf{e_j} \in O_\alpha$. Now $|\mathbf{z}| = |\mathbf{x}| - 1 = |\mathbf{y}|$, and $\mathbf{z}$ is in O_α, $\mathbf{y}$ in a later orthant, so $\mathbf{z} \prec \mathbf{y}$, and $\mathbf{z} \in C$—a contradiction.

Thus

$$(N(C) \cap O_\alpha) \backslash N_{O_\alpha}(C \cap O_\alpha) = \emptyset \text{ or } \{\text{1st point of } O_\alpha\},$$

which proves the lemma. □

Before proceeding, we make explicit the isomorphism between each orthant and the grid.

For $\mathbf{x} \in O_\alpha$, let $\phi_\alpha(\mathbf{x})$ be the point of $[h]^n$ defined by

$$\phi_\alpha(\mathbf{x})_i = \left\{ \begin{array}{ll} x_i - 1 & \alpha(i) = +1, \\ -x_i & \alpha(i) = -1. \end{array} \right.$$

Note that ϕ_α is an isomorphism of graphs between O_α and $[h]^n$, and that for $\mathbf{x}$, $\mathbf{y}$ in O_α, $\phi_\alpha(\mathbf{x}) < \phi_\alpha(\mathbf{y})$ in the simplicial order iff $\mathbf{x} \prec \mathbf{y}$ in $\mathbb{Z}_k^n$.

LEMMA 5.2. *Neighborhoods of initial segments of* $(\mathbb{Z}_k^n, \prec)$ *are also initial segments. More precisely, if* X *satisfies* (i), (ii), (iii), *and* (iv) *of (section* 4*), then* $N(X)$ *satisfies*

(i′) $H_r \subseteq N(X) \subseteq H_{r+1}$,

(ii′) *in layer* $r+1$, $N(X)$ *is full in orthants preceding* O_α, *and empty in orthants following* O_α,

(iii′) $N(X) \cap O_\alpha = N_{O_\alpha}(X \cap O_\alpha)$.

Note that (iii′) implies that $N(X) \cap O_\alpha$ is an initial segment of O_α (using ϕ_α and the corresponding result for the grid, given in [2]), and with (i′) and (ii′) that N(X) is an initial segment of $(\mathbb{Z}_k^n, \prec)$, so it suffices to prove the more precise form.

Proof. Assertion (i′) follows from $N(H_s) = H_{s+1}$ if $s \geq 0$, and (iii′) follows from (iv) and Lemma 5.1.

Note that the first point $\mathbf{x}^\beta$ in orthant O_β has

$$x_i^\beta = \begin{cases} 0 & \beta(i) = -1, \\ 1 & \beta(i) = +1, \end{cases}$$

so $|\mathbf{x}^\beta| = \#_+(\beta) = |\{i : \beta(i) = +1\}|$. Also, a neighbor of $\mathbf{x}^\beta$ is either another point in O_β, or the first point of some O_γ, where β and γ differ in only one place.

Now Lemma 5.1 gives that $\mathbf{x} \in O_\beta$ is in $N(X)$ iff it is in $N_{O_\beta}(X \cap O_\beta)$—with the possible exception of the first point in O_β, i.e., $\mathbf{x}^\beta$. Without this qualification, we would have (ii′) following from (ii). The qualification can only affect this when $\mathbf{x}^\beta$ lies in L_{r+1}. From the above it is straightforward to check that in this case $\mathbf{x}^\beta$ does or does not have a neighbor in X as $\beta \prec \alpha$, or $\beta \succ \alpha$. This proves Lemma 5.2. ∎

6. Compression. For $X \subseteq \mathbb{Z}_k^n$ ($n \geq 2$), $1 \leq i \leq n$, and $t \in \mathbb{Z}_k$, the tth *i-slice* of X is

$$X_t^{(i)} = \{\mathbf{x} \in \mathbb{Z}_k^{n-1} : (x_1, \ldots, x_{i-1}, t, x_i, \ldots, x_{n-1}) \in X\},$$

i.e., the set $\{\mathbf{x} \in X : x_i = t\}$, considered as a subset of $\mathbb{Z}_k^{n-1}$.

The *i-compression* $C^{(i)}X$ is defined by

$$(C^{(i)}X)_t^{(i)} = \text{ initial segment of } \mathbb{Z}_k^{n-1} \text{ of size } |X_t^{(i)}|.$$

Thus $|C^{(i)}X| = |X|$.

Now

$$N(X)_t^{(i)} = X_{t-1}^{(i)} \cup N(X_t^{(i)}) \cup X_{t+1}^{(i)},$$

and

$$N(C^{(i)}X)_t^{(i)} = (C^{(i)}X)_{t-1}^{(i)} \cup N((C^{(i)}X)_t^{(i)}) \cup (C^{(i)}X)_{t+1}^{(i)}. \tag{6.1}$$

By definition $|(C^{(i)}X)_s^{(i)}| = |X_s^{(i)}|$, and since $(C^{(i)}X)_t^{(i)}$ is an initial segment of $\mathbb{Z}_k^{n-1}$ the assumption that Theorem 4.1 holds in dimension $n-1$ gives $|N((C^{(i)}X)_t^{(i)})| \leq |N(X_t^{(i)})|$. Also, the three sets in (6.1) are all initial segments and hence nested, so $|N(C^{(i)}X)_t^{(i)}| \leq |N(X)_t^{(i)}|$, and summing over $t \in \mathbb{Z}_k$, we have $|N(C^{(i)}X)| \leq |N(X)|$.

Since the order induced on $\mathbb{Z}_k^{n-1}$, considered as a slice of $\mathbb{Z}_k^n$, is just $\prec$ as defined on $\mathbb{Z}_k^{n-1}$, the operator $C^{(i)}$, if it does anything, replaces points by points earlier in

$(\mathbb{Z}_k^n, \prec)$. Thus the positive integer $w(X) = \sum_{\mathbf{x}\in X}$ (position of $\mathbf{x}$ in $\prec$) decreases. Hence, repeatedly applying $C^{(1)}, \ldots, C^{(n)}$ starting with a set X eventually gives a set CX with

$$|N(CX)| \leq |N(X)|,$$
$$|CX| = |X|,$$

and CX *compressed* in the sense that $C^{(i)}(CX) = CX$ for $i = 1, \ldots, n$, i.e., CX has all its slices in any direction initial segments of $\mathbb{Z}_k^{n-1}$. This permits an important reduction of the problem of proving Theorem 4.1, which we state as a lemma.

LEMMA 6.1. *If Theorem* 4.1 *is false, then there is a counterexample of minimal dimension which is compressed.*

Proof. Suppose X were a counterexample to Theorem 4.1 of minimal dimension. Then CX, having the same size as X but smaller neighborhood, would also be a counterexample. □

7. The neighborhood of a compressed set. Let $X \subseteq \mathbb{Z}_k^n$, where $n \geq 3$, be compressed, and consider the slices of X in the n^{th} direction, $X_t = X_t^{(n)}$.

If $1 \leq r \leq n-1$, $\mathbf{x} \in X$ and $2 \leq x_n \leq h$, then $\mathbf{y} = \mathbf{x} - \mathbf{e_n}$, $\mathbf{x} + \mathbf{e_r} - \mathbf{e_n}$, $\mathbf{x} - \mathbf{e_r} - \mathbf{e_n}$ all precede $\mathbf{x}$. Also, since $n \geq 3$ there is some i, $1 \leq i \leq n$, $i \neq r, n$. Now $\mathbf{x}$ and $\mathbf{y}$ are in the same i-slice, so as X is compressed, $\mathbf{x} \in X$ implies $\mathbf{y} \in X$. This shows that if $\mathbf{x} \in X_t$, $2 \leq t \leq h$, then $\mathbf{x}$ and all its neighbors are in X_{t-1}, i.e.,

$$N(X_t) \subseteq X_{t-1} \quad t = 2, \ldots, h,$$
$$N(X_t) \subseteq X_{t+1} \quad t = -1, \ldots, -h+1, \text{ similarly.}$$

If $\mathbf{x} \in X$ has $x_n = 0$ but $x_r \neq 0$ for some r, then $\mathbf{y} = \mathbf{x} - \sigma(x_r)\mathbf{e_r} + \mathbf{e_n} \prec \mathbf{x}$, as $|\mathbf{y}| = |\mathbf{x}|$, $\sigma(y_n) = +1$, $\sigma(x_n) = -1$. Thus $\mathbf{x} \in X_0 \backslash \{\mathbf{0}\}$ implies $\mathbf{x}$ has a neighbor in X_1, i.e., $X_0 \backslash \{\mathbf{0}\} \subseteq N(X_1)$.

Also, if $i < j$ in $(\mathbb{Z}_k, \prec)$, then $X_j \subseteq X_i$. We can now describe the n-slices of $N(X)$. For $t = 2, \ldots, h$,

$$\begin{aligned} N(X)_t &= N(X_t) \cup X_{t-1} \cup X_{t+1} \\ &= X_{t-1}. \end{aligned}$$

For $t = -1, \ldots, -h+1$,

$$N(X)_t = X_{t+1},$$

similarly. Also,

$$N(X)_0 = N(X_0),$$

as $X_{-1} \subseteq X_1 \subseteq X_0 \subseteq N(X_0)$, and

$$\begin{aligned} N(X)_1 &= N(X_1) \cup X_0 \cup X_2 \\ &= N(X_1), \end{aligned}$$

provided $X_0 \backslash \{\mathbf{0}\} \neq \emptyset$, in which case $X_0 \backslash \{\mathbf{0}\} \subseteq N(X_1)$ implies $X_0 \subseteq N(X_1)$, as $N(X_1)$ is an initial segment.

LEMMA 7.1. *If $n \geq 3$ and $X \subseteq \mathbb{Z}_k^n$ is compressed, then either $|X| \leq 2$, or*

$$|N(X)| = |X| - (|X_{-h+1}| + |X_h|) + |N(X_0)| + |N(X_1)|. \tag{7.1}$$

Proof. From above either $X_0 \subseteq \{\mathbf{0}\}$, whence $|X| \leq 2$, or $N(X)$ consists of the slices $X_{-h+2}, \ldots, X_{h-1}$ moved outwards and $N(X_0)$, $N(X_1)$ added in the middle, giving (7.1). □

8. A symmetry of the torus. In this brief section we describe a symmetry of the torus and deduce some results which will be useful in several places in the proof of Theorem 4.1.

Consider the map $\rho = \rho_n : \mathbb{Z}_k^n \to \mathbb{Z}_k^n$ sending each point $\mathbf{x}$ to its opposite point, given by $\rho(\mathbf{x})_i = x_i + h$. This is an automorphism of $\mathbb{Z}_k^n$ as a graph which reverses $\prec$.

Thus for example,

$$\begin{aligned} & X \text{ is an initial segment} \\ \Leftrightarrow & X^c \text{ is a final segment} \\ \Leftrightarrow & \rho(X^c) \text{ is an initial segment.} \end{aligned}$$

Similarly, since in n dimensions ρ_n maps the slice $x_i = t$ to the slice $x_i = t + h$ by ρ_{n-1},

$$X_t^{(i)} \text{ is an initial segment} \Leftrightarrow \rho_{n-1}((X_t^{(i)})^c) \text{ is an initial segment.}$$
$$C^{(i)}X = X \Leftrightarrow C^{(i)}\rho(X^c) = \rho(X^c).$$

Thus X is compressed if and only if $\rho(X^c)$ is compressed.

9. The middle two slices of compressed sets. Let $X \subseteq \mathbb{Z}_k^n$ ($n \geq 3$) be compressed. We wish to examine the possibilities for X_0 and X_1.

As X_1 is an initial segment of $\mathbb{Z}_k^{n-1}$ we can pick r so that $H_{r-1} \subseteq X_1 \subsetneq H_r$. Let $\mathbf{l}_r$, $\mathbf{l}_{r+1}$ be the last points in $\mathbb{Z}_k^{n-1}$ in layers r, $r+1$, respectively. Then $(\mathbf{l}_r, 1) \prec (\mathbf{l}_{r+1}, 0)$. Also $\mathbf{l}_r$ and $\mathbf{l}_{r+1}$ have a common coordinate (see below). Thus since X is compressed and $(\mathbf{l}_r, 1) \notin X$, we have $(\mathbf{l}_{r+1}, 0) \notin X$, so $\mathbf{l}_{r+1} \notin X_0$ and as X_0 is an initial segment $X_0 \subsetneq H_{r+1}$. Also, from section 7 $X_1 \subseteq X_0$, so $H_{r-1} \subseteq X_0$, and either

$$H_r \subseteq X_0 \subsetneq H_{r+1}, \tag{a}$$

or

$$H_{r-1} \subseteq X_0 \subsetneq H_r. \tag{b}$$

If either X_0 or X_1 is a Hamming ball, then X_0, X_1 are the middle two slices of an initial segment. Suppose that this is not the case. If (a) holds set $X_+ = X_0$, $s = r+1$, $X_- = X_1$, and $t = r$. If (b) holds set $X_+ = X_1$, $s = r$, $X_- = X_0$, and $t = r$. Then X_+, X_- are incomplete in layers s, t, respectively, and X_+ is "too large" X_- "too small" for X_+, X_- to form the middle two slices of an initial segment.

LEMMA 9.1. (i) *X_+ misses some point in the first two orthants of layer s.*

(ii) *X_- contains some point in the last two orthants of layer t.*

Proof. Let $\mathbf{x}$, $\mathbf{y} \in \mathbb{Z}_k^{n-1}$ with $|\mathbf{x}| = s$, $|\mathbf{y}| = t$. If $X_+ = X_i$, $X_- = X_j$ (where $\{i, j\} = \{0, 1\}$), then set $\mathbf{x}' = (x_1, \ldots, x_{n-1}, i)$ and $\mathbf{y}' = (y_1, \ldots, y_{n-1}, j)$. Then in $\mathbb{Z}_k^n$ we have $\mathbf{y}' \prec \mathbf{x}'$. If (a) holds, then $|\mathbf{x}'| = |\mathbf{y}'| = r + 1$, $\sigma(y_n') = +1$, and $\sigma(x_n') = -1$, and if (b) holds then $|\mathbf{y}'| < |\mathbf{x}'|$. Together with the fact that X is compressed in all directions, this proves the following statement.

(†) Let $\mathbf{x}$, $\mathbf{y} \in \mathbb{Z}_k^{n-1}$ with $|\mathbf{x}| = s$, $|\mathbf{y}| = t$. If $\mathbf{x}$ and $\mathbf{y}$ have a common coordinate $x_m = y_m$, and if $\mathbf{x} \in X_+$, then we have $\mathbf{y} \in X_-$.

We look at 1-coordinates.

What is the 1-coordinate of the last point $\mathbf{x}$ in $\mathbb{Z}_k^{n-1}$ in layer t? There are three cases:

(L1) $t \leq (h-1)(n-2)$: the last point is in the last orthant and has minimum possible $|x_1|$, so $x_1 = 0$.

(L2) $(h-1)(n-2) \leq t \leq (h-1)(n-1)$: as above, but minimum possible $|x_1|$ implies $\mathbf{x} = (x_1, -h+1, \ldots, -h+1)$, so $x_1 = -(t-(h-1)(n-2))$.

(L3) $t > (h-1)(n-1)$: there are no points in orthants O_β with $\#_+(\beta) > t-(h-1)(n-1)$, so the last point is the last point in the last O_γ with $\#_+(\gamma) = t-(h-1)(n-1)$, i.e., $\gamma = ++\cdots+--\cdots-$, which has only one point $(h, \ldots, h, -h+1, \ldots, -h+1)$. Thus $x_1 = h$.

Considering the first two orthants in layer s, we have the following.

(F1) $s < n-2$: there are no points in the first two orthants. The first point $\mathbf{y}$ in layer s has $y_1 = 0$.

(F2) If $0 \leq w \leq (h-1)$ and $w + (n-2) \leq s \leq w + h(n-2)$ there is a point $\mathbf{y}$ in the second orthant with $y_1 = -w$.

Since the cases (L1), (L2), and (L3) are exhaustive, we can deduce the following statement.

(‡) If $s = t$ or $t+1$ and $0 \leq s, t \leq h(n-1)$, then in $\mathbb{Z}_k^{n-1}$ the last point in layer t matches 1-coordinate with either the first point, or some point in the first two orthants of layer s.

This proves half of Lemma 9.1, namely (i)—if X_+ contains all points of the first two orthants in layer s, then (as X_+ not a Hamming ball) it contains the first point in layer s. Together with (†) and (‡), this implies that X_- contains the last point of layer t, and hence all of layer t—contradicting X_- not a Hamming ball.

Assertion (ii) follows similarly, or by using the symmetry ρ to transform (‡) to a statement about the first point in layer s and the last point or last two orthants of layer t. □

Now by Lemma 9.1, in layer s X_+ misses some point in the first two orthants, so since X_+ is an initial segment it cannot contain any points in later orthants. Similarly, X_- cannot miss any points in layer t except in the last two orthants. In summary, if X is compressed then moving some number (possibly zero) of points from the first two orthants of X_+ (one of X_0, X_1) to X_- (the other) gives X'_+, X'_- which are the middle two slices of some initial segment.

10. Moving points between slices. We wish to use Lemma 3.1 to give a result in the torus which can be applied to the middle two slices of compressed sets.

We use the map π from the torus to the grid which maps $\mathbf{x}$ to $\phi_\alpha(\mathbf{x})$ (defined in section 5) when $\mathbf{x} \in O_\alpha$. Note that the points added to the neighborhood by $\mathbf{x}$ which lie in O_α are in bijection (via π) with the points added to the neighborhood in the grid by $\pi\mathbf{x}$.

Let A, B, C, D be initial segments of $\mathbb{Z}_k^{n-1}$, $|s-t| \leq 1$ with $|B|-|A| = |D|-|C| \geq 0$,

$$B \backslash A \subseteq \text{first two orthants in layer } s,$$
$$D \backslash C \subseteq \text{last two orthants in layer } t,$$

and either $A = H_{s-1}$ or $D = H_t$.

Now π maps $L_r \cap O_\alpha$ to $L_{r-\#_+(\alpha)}$ of the grid in an order-preserving way, and since the first two orthants of $\mathbb{Z}_k^{n-1}$ have $\#_+ = n-1$, $n-2$, and the last two $\#_+ = 1, 0$,

$$\begin{array}{lcl} \pi : B\backslash A & \mapsto & \text{consecutive points in layers} \quad s-n+1, \quad s-n+2, \\ \pi : D\backslash C & \mapsto & \text{consecutive points in layers} \quad\quad t-1, \quad\quad\quad t, \end{array}$$

and as $n \geq 3$, $t \geq s-1$, we have $t-1 \geq s-n+1$.

Also,

$$A \text{ is a Hamming ball} \Rightarrow \pi(B\backslash A) \text{ starts at start of layer } s-n+1,$$
$$D \text{ is a Hamming ball} \Rightarrow \pi(D\backslash C) \text{ ends at end of layer } t.$$

So π maps $B\backslash A$ and $D\backslash C$ to two consecutive blocks of $|B|-|A|$ points in $[h]^{n-1}$, $\pi(D\backslash C)$ starts not earlier than $\pi(B\backslash A)$ and either $\pi(B\backslash A)$ starts with the first point in a layer, or $\pi(D\backslash C)$ ends with the last point in a layer. These are just the conditions of Lemma 3.1, so $\pi(B\backslash A)$ adds at least as many points to the neighborhood as $\pi(D\backslash C)$, i.e., the points of $B\backslash A$ add at least as many points in their own orthants to the neighborhood as do those of $D\backslash C$.

When can $\mathbf{x}$ add a point $\mathbf{y}$ in a different orthant to the neighborhood? By Lemma 5.1 only when $\mathbf{y}$ is the first point in some O_β, which implies that $\mathbf{x}$ is the first point in some O_α. This cannot happen if O_α is the second last orthant—otherwise $\mathbf{x} = (1,0,\dots,0)$ so $\mathbf{y} = (1,0,\dots,0,1,0,\dots,0)$ and $\mathbf{y}$ has a neighbor $(0,\dots,0,1,0,\dots,0)$ earlier than $\mathbf{x}$. If O_α is the last orthant then $\mathbf{x} = \mathbf{0}$.

Thus we have shown that the points of $B\backslash A$ add at least as many points in their own orthants to the neighborhood as do those of $D\backslash C$, and if we impose $C \neq \emptyset$, so that $\mathbf{0} \notin D\backslash C$, then extra points can be added only by $B\backslash A$, not by $D\backslash C$. We state what we have just proved as a lemma.

LEMMA 10.1. *If A,B,C,D are initial segments of* $\mathbb{Z}_k^{n-1}$, $|s-t| \le 1$ *with* $|B|-|A| = |D|-|C| \ge 0$, $C \neq \emptyset$,

$$B\backslash A \subseteq \textit{first two orthants in layer } s,$$
$$D\backslash C \subseteq \textit{last two orthants in layer } t,$$

and either $A = H_{s-1}$ *or* $D = H_t$, *then*

$$|N(B)| - |N(A)| \ge |N(D)| - |N(C)|.$$

This implies $|N(X_0')| + |N(X_1')| \le |N(X_0)| + |N(X_1)|$—set $A = X_+'$, $B = X_+$, $C = X_-$, $D = X_-'$, noting that if $X_- = \emptyset$, then X_- is a Hamming ball, so $X_0' = X_0$ and $X_1' = X_1$.

11. Constructing a set from two slices. We wish to define a method of constructing a set from two slices which will bear some relation to the original set if we start with the middle two slices of a compressed set, or an initial segment.

The relationship

$$N(X_t) \subseteq \begin{cases} X_{t-1} & t = 2,\dots,h, \\ X_{t+1} & t = -1,\dots,-h+1, \end{cases}$$

for X compressed (established in section 7) suggests a way of doing this: given two slices Y_0, Y_1 we shall define Y_t from the center outwards to be the largest set satisfying the condition corresponding to the above.

For G a graph and X a subset of $V(G)$ the *interior* of X, $int(X)$ is $\{\mathbf{x} \in V(G) : \mathbf{x}$ and all its neighbors lie in $X\}$. Given L_0, L_1 initial segments in $\mathbb{Z}_k^{n-1}$, define $Y = int(L_0, L_1)$ by

$$Y_0 = L_0,\ Y_1 = L_1,$$
$$Y_t = \begin{cases} int(Y_{t-1}) & t = 2,\dots,h, \\ int(Y_{t+1}) & t = -1,\dots,-h+1. \end{cases}$$

Then for X compressed, $X \subseteq int(X_0, X_1)$.

This process also works well with initial segments. To prove this we need a result like Lemma 5.2 for iterated interiors, $int^0 X = X$, $int^{m+1} X = int(int^m X)$. As $int\, \rho(X^c) = \rho(N(X)^c)$, this can be obtained from Lemma 5.2 as follows.

LEMMA 11.1. *Let $X \neq \mathbb{Z}_k^{n-1}$ be an initial segment of $\mathbb{Z}_k^{n-1}$, and O_β some orthant such that*

(i) $H_s \subseteq X \subseteq H_{s+1}$,

(ii) *in layer $s+1$, X is full in orthants preceding O_β and empty in orthants following O_β,*

(iv) $X^c \cap O_\beta \cap L_{s+1} \neq \emptyset$.

Then $H_{s-m} \subseteq int^m(X) \subseteq H_{s-m+1}$, in layer $s-m+1$, $int^m(X)$ is full in orthants preceding O_β and empty in orthants following O_β, $int^m(X) \cap O_\beta = int^m_{O_\beta}(X \cap O_\beta)$, and $int^m(X)$ is an initial segment.

Proof. The case $m = 0$ is trivial. For $m = 1$, apply Lemma 5.2 to $\rho(X^c)$ with $r = (n-1)h - (s+1)$, $\alpha = -\beta$. (Condition (iii) is automatically satisfied as $\rho(X^c)$ is an initial segment). For $m > 1$ use the case $m = 1$ and induction on m.
(There is a difficulty here: condition (iv) is met for $m - 1 = 0$, but may not be met for larger m. However, when it first fails it is because $\#_+(\beta)$ is just too great for O_β to contain points of the layer being considered. Thus the case $m = 1$ can be applied using the next orthant γ with $\#_+(\gamma) < \#_+(\beta)$, for which (iv) will hold. As the orthants between O_β and O_γ contain no points of the relevant layer, the conclusion as stated above still holds. This trick may have to be repeated.) □

LEMMA 11.2. *If C_0, C_1 are initial segments of $\mathbb{Z}_k^{n-1}$, and for some r either*

$$\begin{array}{rccc} \text{(i)} & C_0 = H_r, & & H_{r-1} \subseteq C_1 \subseteq H_r, \\ \textit{or}\ \text{(ii)} & H_r \subseteq C_0 \subseteq H_{r+1}, & & C_1 = H_r, \end{array}$$

then, unless $C_0 = \mathbb{Z}_k^{n-1}$, $C = int(C_0, C_1)$ is an initial segment of $\mathbb{Z}_k^n$.

(Note that this is precisely the form of the middle two n-slices of an initial segment, as in layer r all points with $x_n = 1$ precede all those with $x_n = 0$.)

Proof. Note first that $H_{r-|t|} \subseteq C_t \subseteq H_{r-|t|+1}$ for $t = -h+1, \ldots, h$ (by induction on $|t|$). Thus $H_r \subseteq C \subseteq H_{r+1}$. (1)

If C_0 and C_1 are both Hamming balls, then C is H_{r+1}, H_r or $H_r \cup (L_{r+1} \cap \{\mathbf{x} : \sigma(x_n) = 1\})$, all initial segments. Otherwise let X_0 be whichever of C_0, C_1 is not a Hamming ball, and let $X_i = int^i(X_0)$ for $i = 1, \ldots, h-1$. Let $s = r$ if $X_0 = C_0$, and $s = r-1$ if $X_0 = C_1$, so

$$H_s \subseteq X_0 \subsetneq H_{s+1}.$$

Let O_β be the first orthant of $\mathbb{Z}_k^{n-1}$ such that X_0 is not full in O_β in layer $s+1$. Applying Lemma 11.1 to X_0 shows that X_i is full in orthants before O_β, empty in orthants after O_β (in layer $s-i+1$), and that $X_i \cap O_\beta = int^i_{O_\beta}(X_0 \cap O_\beta)$. (2)

Going back to C: let O_α be the orthant of $\mathbb{Z}_k^n$ corresponding to O_β, with

$$\alpha(n) = \begin{cases} +1 & \text{if } X_0 = C_1, \\ -1 & \text{if } X_0 = C_0. \end{cases}$$

Condition (i) or (ii) of Lemma 11.2 and (2) give that, in layer $r+1$, the set C is full/empty in orthants before/after O_α. (3)

Also from (2), $\phi_\alpha(C \cap O_\alpha)$ has the property that each slice after the first is the interior of the preceding one. This implies that $\phi_\alpha(C \cap O_\alpha)$ is an initial segment of the

grid (if the first slice consists of all points before $\mathbf{x}$ in the simplicial order on $[h]^{n-1}$, then $\phi_\alpha(C \cap O_\alpha)$ consists of all points before $(\mathbf{x}, 0)$ in the simplicial order on $[h]^n$), so $C \cap O_\alpha$ is an initial segment of O_α. (4)

Finally, using (1), (3), and (4) we have that C is an initial segment of $\mathbb{Z}_k^n$. □

12. Proof of Theorem 4.1. We are now ready to combine the results of the previous sections to prove Theorem 4.1 by induction on n.

When $n = 1$ the initial segments of $(\mathbb{Z}_k, \prec)$ are all intervals and so have minimal neighborhoods. The case $n = 2$ can be proved by some straightforward but messy case-checking, and since this provides no insight for the rest of the proof, it is relegated to the appendix. From now on we assume $n \geq 3$, and that the result holds in dimension $n - 1$.

Suppose Theorem 4.1 is false. Then by Lemma 6.1 there is a compressed counterexample X. We may assume that $|X| > 2$ as the initial segments of sizes 0,1,2 clearly have minimal neighborhoods. Let C be the initial segment with $|C| = |X| > 2$. Then C is also compressed, so by Lemma 7.1

$$|N(X)| = |X| - (|X_{-h+1}| + |X_h|) + |N(X_0)| + |N(X_1)|,$$

and

$$|N(C)| = |C| - (|C_{-h+1}| + |C_h|) + |N(C_0)| + |N(C_1)|.$$

Therefore Theorem 4.1 follows if we prove that among compressed sets X of a given size, the initial segment C minimizes $|N(X_0)|+|N(X_1)|$ and maximizes $|X_{-h+1}|+|X_h|$.

The second assertion is equivalent to the initial segment $\rho(C^c)$ minimizing $|Y_0| + |Y_1|$ among compressed sets Y $(= \rho(X^c))$ of the same size, so it suffices to prove that $X \subseteq \mathbb{Z}_k^n$ compressed, C the initial segment with $|C| = |X|$ implies

$$\begin{aligned} |C_0| + |C_1| &\leq |X_0| + |X_1|, \\ |N(C_0)| + |N(C_1)| &\leq |N(X_0)| + |N(X_1)|. \end{aligned} \tag{12.1}$$

(Note that these conditions are exactly (2.1) and (2.2).)

We know from section 10 that we can modify X_0 and X_1 to give X_0', X_1' which are the middle two slices of an initial segment, in such a way that

$$\begin{aligned} |X_0'| + |X_1'| &= |X_0| + |X_1|, \\ |N(X_0')| + |N(X_1')| &\leq |N(X_0)| + |N(X_1)|. \end{aligned} \tag{12.2}$$

We wish to relate X_0', X_1' to C_0, C_1 by using the $int(L_0, L_1)$ operation to generate an initial segment from X_0' and X_1'. Unfortunately, we cannot do this if $X_0' = \mathbb{Z}_k^{n-1}$. We deal with this case first.

If $X_0 \neq \mathbb{Z}_k^{n-1}$ then X_0 needed modifying, so X_0 is not a Hamming ball and $X_0 \subsetneq H_{h(n-1)-1}$. This contradicts $X_0' = \mathbb{Z}_k^{n-1}$ as points are only added in one layer. Thus $X_0 = \mathbb{Z}_k^{n-1}$ also. Now $X_1 = \mathbb{Z}_k^{n-1}$ or $H_{h(n-1)-1}$. In the first case (12.1) trivially holds, so suppose $X_1 = H_{h(n-1)-1}$. Then $X_{-1} \subseteq X_1$ so taking interiors outwards from X_1 and X_{-1}, we have $X_t \subseteq H_{h(n-1)-|t|}$, so $X \subseteq D = H_{h(n-1)}$, an initial segment. Thus $|C| = |X| \leq |D|$, so $C \subseteq D$ and $C_0 \subseteq D_0 = X_0$, $C_1 \subseteq D_1 = X_1$, which implies (12.1).

This leaves the case where $X_0' \neq \mathbb{Z}_k^{n-1}$. Let $X' = int(X_0', X_1')$, an initial segment by Lemma 11.2. We shall compare this with X two slices at a time, using a result like Lemma 10.1, but for interiors.

LEMMA 12.1. *If A, B, C, D are initial segments of $\mathbb{Z}_k^{n-1}$, $|s'-t'| \leq 1$ with $|A|-|B| = |C|-|D| \geq 0$, $C \neq \mathbb{Z}_k^{n-1}$,*

$$A \backslash B \subseteq \textit{last two orthants in layer } s',$$
$$C \backslash D \subseteq \textit{first two orthants in layer } t',$$

and either $A = H_{s'}$ or $D = H_{t'-1}$, then

$$|int(A)| - |int(B)| \geq |int(C)| - |int(D)|.$$

Proof. Apply Lemma 10.1 to $\rho(A^c)$, $\rho(B^c)$, $\rho(C^c)$, $\rho(D^c)$, with $s = (n-1)h - s'$, $t = (n-1)h - t'$. □

To show $|int(X_0', X_1')| \geq |int(X_0, X_1)|$ it suffices to show

$$|int^m(X_-')| + |int^m(X_+')| \geq |int^m X_-| + |int^m X_+|, \tag{12.3}$$

since summing over $m = 0, 1, \ldots, h-1$ gives the result. Now (12.3) holds for $m = 0$, and can be proved by induction on m: Lemma 11.1 implies that $A = int^{m-1}(X_-')$, $B' = int^{m-1}(X_-)$, $C = int^{m-1}(X_+)$, $D = int^{m-1}(X_+')$, satisfy all the conditions of Lemma 12.1 except $|A|-|B'| = |D|-|C|$. However, the induction hypothesis implies that $|A|-|B'| \geq |C|-|D|$, so adding some points to B' to form B, we can apply Lemma 12.1. Since $|int(B')| \geq |int(B)|$, (12.3) follows by induction.

We have now shown that $|X'| \geq |int(X_0, X_1)| \geq |X| = |C|$, so $C \subseteq X'$. Thus

$$|C_0| + |C_1| \leq |X_0'| + |X_1'|,$$
$$|N(C_0)| + |N(C_1)| \leq |N(X_0)'| + |N(X_1)'|,$$

which combined with (12.2) proves (12.1), and hence Theorem 4.1. □

Having proved Theorem 4.1 with its restrictions $4 \leq k_1 \leq \cdots \leq k_n$ and k_i even, it is natural to ask whether these restrictions can be dropped. For the restriction $4 \leq k_1$ the answer may well be yes, though the proof would need modifying (section 9 uses $4 \leq k_1$). In particular, when $k_1 = \cdots = k_n = 2$ Theorem 4.1 becomes precisely Harper's theorem.

The case when the k_i can be odd is rather different—the example $\mathbb{Z}_3^3$ shows that in general there is no ordering which works. In [1], Bollobás and Leader have conjectured that if $k \geq 3$ is odd and $X \subseteq \mathbb{Z}_k^n$ with

$$|X| \geq \max(|\{\mathbf{x} \in \mathbb{Z}_k^n : d(\mathbf{x}, 0) \leq r\}|, |\{\mathbf{x} \in \mathbb{Z}_k^n : d(\mathbf{x}, 0) \geq s\}|),$$

then

$$|N(X)| \geq \min(|\{\mathbf{x} \in \mathbb{Z}_k^n : d(\mathbf{x}, 0) \leq r+1\}|, |\{\mathbf{x} \in \mathbb{Z}_k^n : d(\mathbf{x}, 0) \geq s-1\}|).$$

Any proof of this is likely to need very different techniques from those used here.

Appendix. The case $n = 2$. In the main body of the paper we have proved Theorem 4.1 by induction on n, assuming the case $n = 2$. Here we prove this case, using the results of section 5, section 6 (compression), and section 8 (the symmetry ρ). Unfortunately, there is a lot of case-checking involved, and this is given in a rather dense form, so this section will be less readable than the rest of the paper.

In two dimensions there are many compressed sets which are not very close to initial segments (e.g., the rectangle $[-s, s] \times [-s, s]$), but the boundaries of both compressed sets and initial segments are sufficiently simple that they can be compared (using Wang and Wang's result for $\mathbb{Z}^2$ [4]).

Let $C \subseteq T = (\mathbb{Z}_{k_1} \times \mathbb{Z}_{k_2})$ be an initial segment. When is $\mathbf{x} \in \partial(C)$?

$$\begin{aligned}\mathbf{x} \in \partial(C) &\Leftrightarrow \mathbf{x} \notin C \text{ and } \mathbf{x} \text{ has a neighbor in } C \\ &\Leftrightarrow \mathbf{x} \notin C \text{ and the earliest neighbor of } \mathbf{x} \text{ lies in } C.\end{aligned}$$

If $x_2 \neq 0, 1$ then, for some choice of sign, $\mathbf{y} = \mathbf{x} \pm \mathbf{e_2}$ has $|\mathbf{y}| = |\mathbf{x}| - 1$, and this is the earliest neighbor of $\mathbf{x}$. Thus $\mathbf{x} \in \partial C \Rightarrow x_2 = 0$ or 1, or $\mathbf{x} \pm \mathbf{e_2} \in C$, so ∂C can contain at most two points in each column, and $|\partial C| \leq 2k_1$.

What happens if $|\partial C| = 2k_1$?

In this case ∂C must contain two points in column h_1, so (as $N(C)$ is an initial segment) $(h_1, 1) \in N(C)$, and its earliest neighbor $(h_1 - 1, 1)$ must be in C. The first initial segment to contain this point is $M = H_{h_1 - 1} \cup \{(h_1 - 1, 1)\}$, so $C \supseteq M$. In particular C has at least $k_1 - 1$ points in column 0, so if $k_1 = k_2$ then ∂C has at most one point in column 0, and $|\partial C| \leq 2k_1 - 1$.

Note that for $C \supseteq M$, $|\partial C|$ decreases as $|C|$ increases: $|\partial(C) \cap \text{column } i|$ decreases from 2 to 1 to 0 as $|C \cap \text{column } i|$ increases from $\leq k_2 - 2$ to $k_2 - 1$ to k_2.

Defining $\partial_T(m)$, $\partial_{\mathbb{Z}^2}(m)$ to be the size of the boundary (in T, $\mathbb{Z}^2$, respectively) of the initial segment of T or $\mathbb{Z}^2$ of size m, we have that $\partial_T(m)$ reaches a maximum of $2k_1$ ($k_1 \neq k_2$) or $2k_1 - 1$ ($k_1 = k_2$) at $m = |M|$, and is then nonincreasing.

On the other hand, $\partial_{\mathbb{Z}^2}(m)$ is nondecreasing (see [4]), and for $m \leq |M|$ the initial segments are the same (the wraparound boundary has not yet been reached), so $\partial_{\mathbb{Z}^2}(m) \geq \partial_T(m)$—each boundary point in T corresponds to at least one in $\mathbb{Z}^2$.

Combining these statements we have

$$\partial_{\mathbb{Z}^2}(m) \geq \partial_T(m) \text{ for all } m. \tag{A.1}$$

From now on we suppose that the $n = 2$ case of Theorem 4.1 is false, and hence that there is a compressed counterexample $X \subseteq T$ (by Lemma 6.1). C will be the initial segment of size $|X|$.

We shall say that a row is *full* (*respectively, almost full*) if X contains k_1 (respectively, $k_1 - 1$) points of the row; similarly for a column, but with k_1 replaced by k_2. We now describe four cases one of which must hold, and deduce a contradiction in each. This will prove the $n = 2$ case of Theorem 4.1.

Case 1. X has no row or column full or almost full.

As X is compressed, $X \subseteq [-h_1 + 2, h_1 - 1] \times [-h_2 + 2, h_2 - 1]$, so considering X as a subset of $\mathbb{Z}^2$, $\partial_{\mathbb{Z}^2}(X) \subseteq [-h_1 + 1, h_1] \times [-h_2 + 1, h_2]$, and in the torus no points of $\partial_{\mathbb{Z}^2}(X)$ are identified.

Setting $m = |X|$ we have

$$\begin{aligned}|\partial_T(X)| &= |\partial_{\mathbb{Z}^2}(X)| \\ &\geq \partial_{\mathbb{Z}^2}(m) \text{ (isoperimetric inequality for } \mathbb{Z}^2\text{—see [4])} \\ &\geq \partial_T(m) \text{ by (A.1) above.}\end{aligned}$$

Therefore X is not a counterexample, contradicting our assumption.

Case 2. X has both a full row and a full column.

As X is compressed, $\{\mathbf{x} : x_1 = 0 \text{ or } x_2 = 0\} \subseteq X$. Therefore $\rho(X^c) \subseteq [-h_1 + 1, h_1 - 1] \times [-h_2 + 1, h_2 - 1]$.

Let $X' = \rho(X^c)$, and let $C' = \rho(C^c)$—the initial segment of size $|X'|$. Then by assumption $|N(X)| < |N(C)|$, so as $int(X^c) = (N(X))^c$, $|int\, X^c| > |int\, C^c|$. Hence, as ρ is an automorphism of the torus, $|int\, X'| > |int\, C'|$.

Let Y be the initial segment of size $|int\, X'|$. Then $|Y| > |int\, C'|$, so

$$\begin{aligned} Y \not\subseteq int\, C' &\Rightarrow N(Y) \not\subseteq C' \text{ (by definition of } int) \\ &\Rightarrow |N(Y)| > |C'|, \end{aligned}$$

as $N(Y)$ and C' are both initial segments, and are hence nested.

Also, by the argument of case 1, as $int\, X' \subseteq [-h_1+2, h_1-2] \times [-h_2+2, h_2-2]$, $int\, X'$ cannot be a counterexample, so $|N(int\, X')| \geq |N(Y)|$ and

$$|C'| = |X'| \geq |N(int\, X')| \geq |N(Y)| > |C'|,$$

a contradiction.

Before proceeding we make a remark.

If $k_1 \neq k_2$, then, as X is a counterexample, $|\partial X| \leq 2k_1 - 1 \leq 2k_2 - 5$. Suppose X has a column which is full or almost full. Then if no rows are full or almost full, ∂X contains at least one point in row k_2 and at least two points in all other rows, so $|\partial X| \geq 2k_2 - 1$. Thus X must have enough rows full or almost full to "lose" at least four neighbors, and in particular either row 0 is full, or at least four rows are almost full. In the second case, rows 0,1 and -1 are almost full (using X compressed), and we can move a point of X to $(h_1, 0)$ without increasing the neighborhood, to give another compressed counterexample with a full row.

Case 3. X has a full row or column.

Without loss of generality, X has a full row (reflect if $k_1 = k_2$, and use the remark above if $k_1 \neq k_2$). As X is compressed, row 0 is full.

Now if no columns are full or almost full, $|\partial X| \geq 2k_1$ (at least two points in each column), and X is not a counterexample. If column 0 is full we are done by case 2. Thus there are no full columns. How many almost full columns can there be?

If more than two columns are almost full we are done, since as columns 0,1,-1 are almost full we can move a point to fill column 0, reducing to case 2. Thus the number of almost full columns is either one or two.

If exactly one column is almost full, then $|\partial X| \geq 2k_1 - 1$ so as X is a counterexample, we have $k_1 \neq k_2$, $|\partial X| = 2k_1 - 1$ and X has exactly two boundary points in columns other than 0. Working outwards from the center, this means that no column can contain more than two points fewer than the previous column, so $X \supseteq H_{h_2-1}$. Hence, $|C| = |X| \geq |H_{h_2-1}|$ so $C \supseteq H_{h_2-1}$ which implies $|\partial C| \leq 2k_1 - 1$ (as C has column 0 at least almost full), so X is not a counterexample.

If exactly two columns are almost full, then working outwards from the center we can have at most one step where a column has three fewer points than the previous column. If we have such a step then $|\partial X| \geq 2k_1 - 1$, but we always have $|\partial C| \leq 2k_1$, and hence $|\partial X| \leq 2k_1 - 1$ for a counterexample X, so in fact $|\partial X| = 2k_1 - 1$. The smallest possibility for X has the step between columns 1 and 2 (or 0 and -1), but in this case certainly $|X| \geq |H_{h_2-1}|$. C thus contains H_{h_2-1}, so $|\partial C| \leq 2k_1 - 1$. If we do not have such a step, then $X = H_{h_2-1} \cup (L_{h_2} \cap \{\mathbf{x} : x_1 > 0\})$ so $C = H_{h_2-1} \cup (L_{h_2} \cap \{\mathbf{x} : x_2 > 0\})$ (the initial segment of the same size) and $|\partial C| = 2k_1 - 2 = |\partial X|$.

In neither of these cases do we have $|\partial X| < |\partial C|$, which contradicts our assumption that X is a counterexample.

Case 4. X has an almost full row or column, but no full row or column.

We consider first the case $k_1 \neq k_2$.

By the remark above, we may assume that column 0 is not almost full, and, hence, that row 0 is almost full. Thus $|\partial X| \geq 2k_1 - 1$, so as X is a counterexample $|\partial X| = 2k_1 - 1$ and X has exactly one neighbor in column h_1 and exactly two in each other column. Working inwards from columns $\pm(h_1 - 1)$ this time, we have that $X \subseteq H_{h_1-1}$. Thus $C \subseteq H_{h_1-1}$ and $|\partial C| \leq |\partial H_{h_1-1}|$ (as until columns become almost full the boundary of an initial segment increases with size, and as $h_1 < h_2$), so $|\partial C| \leq 2k_1 - 1 = |\partial X|$, a contradiction.

We now consider the case $k_1 = k_2 = k$.

Without loss of generality row 0 is almost full. We need $|\partial X| < 2k-1$ for X to be a counterexample, so (counting the number of points of ∂X in each column) column 0 must be almost full as well. If there are no more almost full rows or columns, then we must have $|\partial X| = 2k - 2$ and working outwards from columns 0 and 1 each column must have at most two fewer points than the previous. As columns $\pm(h - 1)$ can each contain only one point, each column must have exactly two fewer points than the previous, so $X = H_{h-1}$, an initial segment and certainly not a counterexample.

If X has another almost full row or column, without loss of generality row 1, then as ∂X has at least two points in column h, X must have another almost full column, say column 1. Now if column (or row) -1 is also almost full, we can move a point as before to create a full column (or row), reducing to case 3. Thus $|\partial X| = 2k - 2$, and ∂X has one point in each of columns 0,1 and exactly two points in each other column. Now for $i = 2, \ldots, h - 1$ column i has at most two fewer points than column $i - 1$, so column $h - 1$ has at least $(k - 1 - 2(h - 2)) = 3$ points, and ∂X has at least three points in column h. This contradicts ∂X having exactly two points in columns other than 0,1.

This completes the proof of the $n = 2$ case of Theorem 4.1.

Acknowledgment. The author would like to thank Dr. B. Bollobás for many detailed suggestions which improved the presentation of this paper.

REFERENCES

[1] B. Bollobás and I. Leader, *An isoperimetric inequality on the discrete torus*, SIAM J. Discrete Math., 3 (1990), pp. 32–37.

[2] B. Bollobás and I. Leader, *Compressions and isoperimetric inequalities*, J. Combin. Theory, 56 (1991), pp. 47–62.

[3] L. H. Harper, *Optimal numberings and isoperimetric problems on graphs*, J. Combin. Theory, 1 (1966), pp. 385–393.

[4] D.-L. Wang and P. Wang, *Discrete isoperimetric problems*, SIAM J. Appl. Math., 32 (1977), pp. 860–870.

[5] D.-L. Wang and P. Wang, *Extremal configurations on a discrete torus and a generalization of the generalized Macaulay theorem*, SIAM J. Appl. Math., 33 (1977), pp. 55–59.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 128–134, February 1998

009

COMPETITION GRAPHS OF HAMILTONIAN DIGRAPHS*

DAVID R. GUICHARD†

Abstract. K. F. Fraughnaugh et al. proved that a graph G is the competition graph of a hamiltonian digraph possibly having loops if and only if G has an edge clique cover $\mathcal{C} = \{C_1, \ldots, C_n\}$ that has a system of distinct representatives. [*SIAM J. Discrete Math.*, 8 (1995), pp. 179–185]. We settle a question left open by their work, by showing that the words "possibly having loops" may be removed.

Key words. competition graph, hamiltonian digraph

AMS subject classification. 05C15

PII. S089548019629735X

1. Introduction. Suppose that D is a digraph. The **competition graph** or **conflict graph** $C(D)$ has the same vertex set as D, and an edge $\{u, v\}$, $u \neq v$, if there is a vertex w such that (u, w) and (v, w) are arcs in D. Competition graphs are useful in the study of such diverse systems as food webs and radio networks and have received substantial attention over the past fifteen years. Characterizations of competition graphs for a variety of classes of digraphs have been reported over the years; recently, K. F. Fraughnaugh et al. [2] have given characterizations for competition graphs of strongly connected digraphs and hamiltonian digraphs. Here we improve their characterizations of competition graphs of hamiltonian digraphs. (See the same paper for references to other characterizations.)

2. Hamiltonian digraphs. All digraphs are loopless and contain no multiple edges unless otherwise indicated. We use **circuit** to mean a directed cycle in a digraph. Fraughnaugh [2] contains the following two characterizations.

THEOREM 1. *A graph G on n vertices is the competition graph of a hamiltonian digraph if and only if G has an edge clique cover $\{C_1, \ldots, C_n\}$, with a system of distinct representatives $\{v_1, \ldots, v_n\}$ such that $v_i \notin C_{i-1}$. (Subscript arithmetic "wraps around," i.e., $C_0 = C_n$.)*

THEOREM 2. *A graph G on n vertices is the competition graph of a hamiltonian digraph, possibly with loops, if and only if G has an edge clique cover $\{C_1, \ldots, C_n\}$ with a system of distinct representatives.*

We show that these characterizations can be "combined" as follows.

THEOREM 3. *A graph G on $n \geq 3$ vertices is the competition graph of a hamiltonian digraph if and only if G has an edge clique cover $\{C_1, \ldots, C_n\}$ with a system of distinct representatives.*

Proof. One direction follows immediately from Theorem 1. The other direction follows if we can show that whenever G has an edge clique cover $\{C_1, \ldots, C_n\}$ with a system of distinct representatives, it has one satisfying the additional property of Theorem 1, namely, that $v_i \notin C_{i-1}$. We show precisely this in Theorem 6. Note that

*Received by the editors January 17, 1996; accepted for publication (in revised form) November 22, 1996.

http://www.siam.org/journals/sidma/11-1/29735.html

†Department of Mathematics and Computer Science, Whitman College, Walla Walla, WA 99362 (guichard@whitman.edu).

K_2 has an edge clique cover with a system of distinct representatives but is not the competition graph of a hamiltonian digraph. □

Suppose that G is a graph on n vertices and $\mathcal{C} = \{C_1, \ldots, C_n\}$ is a clique cover of G with a system of distinct representatives $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. The clique graph $CG(\mathcal{C}, \mathcal{V})$ is the digraph whose vertices are the cliques, with arc (C_i, C_j) if and only if $v_j \notin C_i$. To complete the proof of Theorem 3, we want to show that if G has a clique cover with a system of distinct representatives, then it has one whose clique graph is hamiltonian. (For if the clique graph is hamiltonian, we may renumber the cliques, if necessary, so that $(C_1, C_2, \ldots, C_n)$ is a circuit. By definition of the clique graph, this implies the "extra condition" of Theorem 1.)

Recall the well-known theorem of Ghouila-Houri.

THEOREM 4. *Suppose G is a strongly connected digraph without loops or multiple arcs. If $d(v) = d^+(v) + d^-(v) \geq n$ for every vertex v, then G is hamiltonian.*

We will use this in two ways: in some special cases, we will be able to invoke Theorem 4 directly. For the rest, when the hypotheses of the theorem are not quite met, we give a proof that is much like the proof of Theorem 4, using some properties of the clique graph to make up for the failed hypotheses.

We will need the following technical lemma (also used in the proof of Theorem 4).

LEMMA 5. *Consider a circuit with m vertices, each colored either red or blue. Suppose the circuit contains exactly $n_r \geq 1$ red vertices and $n_b \geq 1$ sequences of q consecutive blue vertices. Then $n_r + n_b \leq m - q + 1$.*

This lemma, and the proof of Theorem 4 that we use, are from Berge [1].

DEFINITION. *If C is a graph, let $|C|$ denote the number of vertices in C. The* **size** *of a set of cliques $\mathcal{C}$ is $\sum_{C \in \mathcal{C}} |C|$.*

Remark. When C is a set, not a graph, we use $|C|$ in the usual sense to mean the number of elements in the set C.

DEFINITION. *If $A = (V_A, E_A)$ and $B = (V_B, E_B)$ are subgraphs of G, by $A \cup B$ we mean the subgraph with vertex set $V_A \cup V_B$ and edge set $E_A \cup E_B$.*

THEOREM 6. *If G is a graph on $n \geq 3$ vertices and has an edge clique cover $\{C_1, \ldots, C_n\}$ with a system of distinct representatives, then it has one whose clique graph is hamiltonian.*

Proof. If G is a complete graph, the theorem is easy. For other G, we prove by induction on the number of vertices that if G has a clique cover with a system of distinct representatives, then among all such clique covers there is one of minimum size whose clique graph is hamiltonian. The theorem is easy to prove for $n = 3$, so suppose $n \geq 4$.

If $\mathcal{C}$ is a clique cover with a system of distinct representatives, let $k(\mathcal{C}) \geq 0$ be the largest integer strictly less than n such that some collection $\mathcal{A}$ of $k(\mathcal{C})$ of the cliques has $|\mathcal{A}| = |\bigcup \mathcal{A}|$. (Recall that Hall's marriage principle says that $|\mathcal{A}| \leq |\bigcup \mathcal{A}|$ for all $\mathcal{A} \subseteq \mathcal{C}$. Here and subsequently, we use $\bigcup \mathcal{A}$ to mean $\bigcup_{C \in \mathcal{A}} C$.)

Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ be a clique cover of G of minimum size with a system of distinct representatives $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, for which $k = k(\mathcal{C})$ is as large as possible. We may assume that $k = |\bigcup_{i=1}^{k} C_i|$.

Unless $k = n - 1$ and C_n is a singleton, the clique cover has the following "minimality property," henceforth (MP): if $v \in C_i$, $i > k$, then there is an edge $\{v, w\}$ in C_i that is in no other clique, for if not, then v could be removed from C_i to form a clique cover with a system of distinct representatives of smaller size. (Note that C_i cannot be a singleton, by definition of k.)

Remark. If every edge $\{v, w\}$ in C_i is in some other clique, then it is clear that removing v from C_i leaves a clique cover. It is perhaps not obvious that this new cover has a system of distinct representatives. (Actually, it is clear if $k = n - 1$, so we may assume that $k < n - 1$.) For a contradiction, suppose it doesn't; by Hall's marriage principle, C_i must be in some set of cliques $\mathcal{A}$ such that $|\mathcal{A}| = |\bigcup \mathcal{A}|$, and C_i is the only clique in $\mathcal{A}$ that contains v. If $v \in \bigcup_{j=1}^{k} C_j$, then $v \neq v_i$, so $\mathcal{A}$ with C_i replaced by $C_i \backslash \{v\}$ has a system of distinct representatives (inherited from $\mathcal{V}$), contradicting the definition of $\mathcal{A}$. Hence, $v \notin \bigcup_{j=1}^{k} C_j$. Now there are two cases: (1) If $\mathcal{A} \supseteq \{C_{k+1}, \ldots, C_n\}$, then $\mathcal{B} = \{C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_n\}$ satisfies $|\mathcal{B}| = |\bigcup \mathcal{B}|$, contradicting the definition of k. (2) Otherwise, $\mathcal{B} = \mathcal{A} \cup \{C_1, \ldots, C_k\}$ satisfies $|\mathcal{B}| = |\bigcup \mathcal{B}|$, contradicting the definition of k. To see that $|\mathcal{B}| = |\bigcup \mathcal{B}|$, pick a system of distinct representatives for $\mathcal{A} \cup \{C_1, \ldots, C_k\}$. If there is a $v \in C_j$, $j \leq k$ that is not one of the representatives, then $k < |\bigcup_{j=1}^{k} C_j|$, contradicting the definition of k. If there is a $v \in C_j$, $C_j \in \mathcal{A}$ that is not one of the representatives, then $|\mathcal{A}| < |\bigcup \mathcal{A}|$, contradicting the definition of $\mathcal{A}$.

Here is an outline of the rest of the proof.

I. Establish some properties of $C_1, \ldots, C_k$ and of arcs between these cliques and $C_{k+1}, \ldots, C_n$.
II. Show that $d(C) \geq n$ for $C \in \{C_{k+1}, \ldots, C_n\}$.
III. Show that $CG(\mathcal{C}, \mathcal{V})$ is strongly connected.
IV. Use Theorem 4 to show that $CG(\mathcal{C}, \mathcal{V})$ is hamiltonian in some special cases.
V. Show that $CG(\mathcal{C}, \mathcal{V})$ is hamiltonian using methods similar to the proof of Theorem 4.

I. Some properties of $C_1, \ldots, C_k$.

We establish some properties of $C_1, \ldots, C_k$, in some cases by replacing $\mathcal{C}$ by a different clique cover.

If $k > 0$, the cliques $C_1, \ldots, C_k$ form a clique cover with a system of distinct representatives for a smaller graph G', namely, the union of the cliques $C_1, \ldots, C_k$. If G' is not a complete graph, then, by the induction hypothesis, we may replace $C_1, \ldots, C_k$ by a clique cover of G', with a system of distinct representatives, that has the same size as $C_1, \ldots, C_k$, and whose clique graph is hamiltonian. For convenience, call the new cliques $C_1, \ldots, C_k$ as well.

If not all of $C_1, \ldots, C_k$ are singletons, every clique C_g, $g > k$ must have an arc (in $CG(\mathcal{C}, \mathcal{V})$) to some C_i, $i \leq k$. If not, C_g contains all of $v_1, \ldots, v_k$, so we could replace the cliques $C_1, \ldots, C_k$ by singletons to get a smaller clique cover, which is a contradiction.

If G' is a complete graph and $k \geq 2$, the cliques $C_1, \ldots, C_k$ must consist of $k - 1$ singletons and a copy of K_k, because $\mathcal{C}$ was chosen to have minimum possible size. To see this, suppose that no C_i, $i \leq k$ is K_k. Then each v in G' must be in at least two of the C_i for $i \leq k$, and so the size of the cover $C_1, \ldots, C_k$ is at least $2k$, while the size of $k - 1$ singletons and of a K_k is $2k - 1$. We may assume that $C_1, \ldots, C_{k-1}$ are the singletons. We also may assume that some C_g, $g > k$ does not contain C_1, by the previous paragraph, and by some renumbering of the vertices of G' if necessary. To see this, note that if every C_g, $g > k$ contains all of $C_1, C_2, \ldots, C_{k-1}$, then no C_g, $g > k$ contains v_k, the representative of C_k. Replacing C_1 by $\{v_k\}$ and renumbering, we get a set of cliques with the desired property. Finally, we may assume that some C_g, $g > k$, has an arc to $C_k \cong K_k$. For if not, then every C_g, $g > k$ contains all of $v_2, \ldots, v_k$ and does not contain v_1. Then, if $k > 2$, we may replace $C_1, \ldots, C_k$

by $\{v_1, v_2\}$, $\{v_1, v_3\}$, ..., $\{v_1, v_k\}$, $\{v_2\}$. Together with $C_{k+1}, \ldots, C_n$, these form a clique cover with a system of distinct representatives of the same size as the original, so we may use this clique cover instead of the original. For convenience, name this new clique cover $\mathcal{C}$, and name the new cliques $C_1, C_2, \ldots, C_n$. If $k = 2$, then we can replace $C_1 = \{v_1\}$ and $C_2 = \{v_1, v_2\}$ by $C_1' = \{v_2\}$ and $C_2' = \{v_1, v_2\}$, and then $\mathcal{A} = \{C_1', C_3, \ldots, C_n\}$ has the property that $|\mathcal{A}| = |\bigcup \mathcal{A}|$, which is a contradiction, by the definition of k.

If $C_1, \ldots, C_k$ are all singletons and every C_g, $g > k$ contains all vertices of G', let v be a vertex of G that is not in all of the cliques C_g, $g > k$. (If every vertex is in every C_g, $g > k$, then G is K_n, contrary to assumption.) Let C be the clique represented by v. Replace the singleton $\{v_1\}$ by $\{v\}$ in $\mathcal{C}$. This new collection of cliques (still called $\mathcal{C}$) is still a clique cover with a system of distinct representatives and has minimum size. Choosing g so that $v \notin C_g$, (C_g, C_1) is an arc of $CG(\mathcal{C}, \mathcal{V})$.

II. $d(C) \geq n$ for $C \in \{C_{k+1}, \ldots, C_n\}$.

Actually, we show that either the theorem is true for G or $d(C) \geq n$ for $C \in \{C_{k+1}, \ldots, C_n\}$.

Consider a clique $C = C_i$, $i > k$. In the clique graph, $d^+(C) = n - |C|$ and

$$\begin{aligned} d^-(C) &= \text{the number of cliques not containing } v_i \\ &= n - \text{the number of cliques containing } v_i \\ &\geq n - (1 + n - |C|) = |C| - 1 \end{aligned}$$

by (MP). Thus, $d(C) \geq n - 1$.

If for some $C = C_i$, $d(C) = n - 1$, then the number of cliques containing v_i is exactly $1 + n - |C|$. (Note that $|C| \geq 2$. If not, then $k = n - 1$ and $i = n$, so $d^-(C) = n - 1$.) Let $j = n - |C|$. Let the cliques other than C that contain v_i be $A_1, \ldots, A_j$. By (MP), there are vertices $w_1, \ldots, w_j$ such that $A_l \backslash C = \{w_l\}$ for all l. Now we consider the cliques other than C and $A_1, \ldots, A_j$. Note that any clique contained in C must be a singleton, since $\mathcal{C}$ was chosen to have minimum size.

Suppose there is a clique D that contains more than one of $\{w_1, \ldots, w_j\}$; without loss of generality, say $\{w_1, \ldots, w_t\} \subseteq D$. Remove v_i from $A_1, \ldots, A_t$, and add v_i to D. This new collection of cliques still covers G, has a system of distinct representatives, and has smaller size than $\mathcal{C}$, which is a contradiction.

Suppose that D is a clique (not one of $A_1, \ldots, A_j$) containing one of $\{w_1, \ldots, w_j\}$, say, w_s, and D is not a singleton. If we replace A_s by $D \cup A_s$ and D by its representative, we get a clique cover of G with a system of distinct representatives and the same size as $\mathcal{C}$. If we do this for each such D, we produce a clique cover, still called $\mathcal{C}$ for convenience, consisting of $C, A_1, \ldots, A_j$ and $|C| - 1 \geq 1$ singletons. Moreover, we may assume that each A_m is represented by w_m, and all of the singletons $\{x_1, \ldots, x_{|C|-1}\}$ are in C.

Suppose that one of the singletons, without loss of generality, x_1, is not in some A, without loss of generality, A_j. Then C, A_1, ..., A_j, x_1,..., $x_{|C|-1}$ is a hamilton circuit in $CG(\mathcal{C}, \mathcal{V})$.

Otherwise, suppose that every singleton is in every clique A. Split A_j into two cliques. One, still called A_j for convenience, is $A_j \backslash \{x_1\}$. The other, A_{j+1}, contains w_j and x_1. Remove the singleton $\{x_1\}$ from $\mathcal{C}$. If we let x_1 represent A_{j+1}, then this new collection of cliques is still a cover, still has a system of distinct representatives, and has the same size as $\mathcal{C}$. In the clique graph of this new collection, C, $A_1, \ldots, A_j$, A_{j+1},

$x_2, \ldots, x_{|C|-1}$ is a hamilton circuit. (Note that if $|C| = 2$, there are no singletons left, but C, $A_1, \ldots, A_j$, A_{j+1} is a hamilton circuit, since $v_i \notin A_{j+1}$.)

Thus, we may assume that $d(C_i) \geq n$, $i > k$.

III. $CG(\mathcal{C}, \mathcal{V})$ is strongly connected.

Consider two distinct cliques, C_i and C_j.

If $i \leq k$ and $j > k$, then by definition of k, (C_i, C_j) is an arc.

If $i, j > k$ and (C_i, C_j) is not an arc, then $v_j \in C_i$. Let $v \in C_j$ be such that $\{v_j, v\}$ is in no clique other than C_j. Let C be the clique represented by v, so $v_j \notin C$. Then (C_i, C) and (C, C_j) are arcs.

Suppose $i, j \leq k$. If G' is not a complete graph, then $C_1, \ldots, C_k$ form a circuit, so there is a path from C_i to C_j.

If G' is the complete graph, the cliques $C_1, \ldots, C_k$ must consist of $k-1$ singletons and a copy of K_k. From (I) we know that $C_1, \ldots, C_{k-1}$ are singletons and that for some $g > k$, (C_g, C_1) is an arc. Taken in order, $C_1, \ldots, C_k$ form a path, so if $i < j$, there is a path from C_i to C_j. If $i > j$, we may use the path $C_i, C_g, C_1, \ldots, C_j$.

Finally, suppose $i > k$ and $j \leq k$. If G' is not a complete graph and $C_1, \ldots, C_k$ are not all singletons, then (C_i, C_l) is an arc for some $l \leq k$, and since $C_1, \ldots, C_k$ form a circuit, there is a path from C_l to C_j. If all of $C_1, \ldots, C_k$ are singletons, there is a $g > k$ and an $l \leq k$ such that (C_g, C_l) is an arc; since there is a path from C_i to C_g and from C_l to C_j, there is a path from C_i to C_j. If G' is a complete graph, we know there is a $g > k$ such that (C_g, C_1) is an arc; since there are paths from C_i to C_g and C_1 to C_j, we are done.

IV. Special cases.

Case 1. $k = 0$.

$CG(\mathcal{C}, \mathcal{V})$ satisfies the hypotheses of Theorem 4, so it is hamiltonian.

Case 2. $k = 1$.

By definition of k, this means that C_1 is a singleton, so $d^+(C_1) = n-1$. Previously, we had guaranteed that there is some $g > k$ such that (C_g, C_1) is an arc, so $d^-(C_1) \geq 1$. Now $CG(\mathcal{C}, \mathcal{V})$ satisfies the hypotheses of Theorem 4, so it is hamiltonian.

Case 3. All of $C_1, \ldots, C_k$ are singletons, and $k > 1$.

If $i \leq k$, then $d^+(C_i) = n - 1$ and $d^-(C_i) \geq 1$ because each of the other singletons has an arc to C_i. Again, $CG(\mathcal{C}, \mathcal{V})$ satisfies the hypotheses of Theorem 4, so it is hamiltonian.

Case 4. G' is a complete graph.

Recall that this means that C_1, $C_2, \ldots, C_{k-1}$ are all singletons, and $C_k \cong K_k$. For $i \leq k-1$, $d^+(C_i) = n - 1$. For $2 \leq i \leq k-1$, (C_{i-1}, C_i) is an arc, so $d^-(C_i) \geq 1$. By (I), we know that there is some $g > k$ such that (C_g, C_1) is an arc, so $d^-(C_1) \geq 1$. Thus, $d(C_i) \geq n$ for $i \leq k-1$.

$d^+(C_k) = n-k$ and $d^-(C_k) \geq (k-1)+1 = k$. The "$k-1$" is due to $C_1, \ldots, C_{k-1}$. By the discussion in (I), there is some $g > k$ such that (C_g, C_k) is an arc, which explains the "+1." Thus, $CG(\mathcal{C}, \mathcal{V})$ satisfies the hypotheses of Theorem 4, so it is hamiltonian.

V. The rest of the story.

Now we may assume that $C_1, \ldots, C_k$ form a circuit, and that $k \geq 2$. Also, we may assume that not all of $C_1, \ldots, C_k$ are singletons, so that for all $g > k$ there is some $i \leq k$

such that (C_g, C_i) is an arc. Following Berge [1], we let $\Gamma^+(v) = \{w \mid (v,w) \text{ is an arc}\}$ and $\Gamma^-(v) = \{w \mid (w,v) \text{ is an arc}\}$.

Let X_0 be the longest circuit in $CG(\mathcal{C}, \mathcal{V})$ that incorporates all of $C_1, \ldots, C_k$ in that order (but not necessarily consecutively). Let m be the length of X_0; $m \geq 2$. Denote the vertices in X_0 by $V_0, V_1, \ldots, V_{m-1}$ (in order around the circuit). If $m = n$, we are done; for a contradiction, suppose that $m < n$. Let $X_1, \ldots, X_p$ be the strongly connected components of $CG(\mathcal{C}) - X_0$.

CLAIM. *X_1 contains a circuit of length $|X_1| = q$.*

Suppose that $V \in X_1$. For all l, $V_l \in \Gamma^-(V)$ implies $V_{l+1} \notin \Gamma^+(V)$, for otherwise X_0 could be lengthened. (Note: $V_m = V_0$.) Hence,

$$|\Gamma^-(V) \cap X_0| \leq |X_0| - |\Gamma^+(V) \cap X_0|.$$

For $W \in X_j$, $j \neq 0, 1$, $W \in \Gamma^-(V)$ implies $W \notin \Gamma^+(V)$ because X_j is a strongly connected component different from X_1. Hence, for $j \neq 0, 1$,

$$|\Gamma^-(V) \cap X_j| \leq |X_j| - |\Gamma^+(V) \cap X_j|.$$

Since $d(V) \geq n$,

$$d(V) = \sum_{j=0}^{p} \Big(|\Gamma^-(V) \cap X_j| + |\Gamma^+(V) \cap X_j| \Big) \geq \sum_{j=0}^{p} |X_j| = n,$$

and so

$$|\Gamma^-(V) \cap X_1| + |\Gamma^+(V) \cap X_1| - |X_1| \geq - \sum_{j \neq 1} \Big(|\Gamma^-(V) \cap X_j| + |\Gamma^+(V) \cap X_j| - |X_j| \Big) \geq 0.$$

By Theorem 4, X_1 is hamiltonian. Denote the vertices of X_1 by $W_0, W_1, \ldots, W_{q-1}$, in order around the circuit. This proves the claim.

CLAIM. *$q < m$.*

If not, we can form a circuit by inserting $C_1, \ldots, C_k$ in order into X_1. (Pick any $V \in X_1$; then (V, C_i) is an arc for some $i \leq k$, and $C_i, C_{i+1}, \ldots, C_k, C_1, \ldots, C_{i-1}$ may be inserted immediately after V in X_1.) This forms a circuit containing $C_1, \ldots, C_k$ that is longer than X_0, which is a contradiction. This proves the claim.

CLAIM. *Suppose that $V_i \in \Gamma^-(W_s)$. Then $V_{i+1}, \ldots, V_{i+q}$ are not in $\Gamma^+(W_{s-1})$. (All subscript arithmetic wraps around as appropriate.)*

For if $V_{i+j} \in \Gamma^+(W_{s-1})$, the circuit $V_0, \ldots, V_i, W_s, W_{s+1}, \ldots, W_{s-1}, V_{i+j}, \ldots, V_{m-1}$ is longer than X_0. Suppose some of $V_{i+1}, \ldots, V_{i+j-1}$ are in $\{C_1, \ldots, C_k\}$; by definition of X_0, these vertices must be $C_g, C_{g+1}, \ldots, C_{g+h}$, for some g and h. (Note that subscript arithmetic here wraps around at k, not n.) These may be inserted in the new circuit immediately following C_{g-1}; if $\{C_g, \ldots, C_{g+1}\} = \{C_1, \ldots, C_k\}$, then $\{C_1, \ldots, C_k\}$ can be inserted anywhere, as in the previous claim. This produces a circuit longer than X_0 that contains all of $C_1, \ldots, C_k$ in order, which is a contradiction. This proves the claim.

Now we show that for each W_s,

$$|\Gamma^-(W_s) \cap X_0| + |\Gamma^+(W_{s-1}) \cap X_0| \leq m - q + 1.$$

Color the vertices of X_0 as follows: if $V_j \in \Gamma^+(W_{s-1})$, color it red; otherwise, blue. We know that both $(\Gamma^-(W_s) \cap X_0)$ and $(\Gamma^+(W_{s-1}) \cap X_0)$ are nonempty (by properties

of $\{C_1, \ldots, C_k\}$), so there is at least one red vertex, and by the preceding paragraph there is at least one sequence of q blue vertices. Thus, by Lemma 5,

$$|\Gamma^-(W_s) \cap X_0| + |\Gamma^+(W_{s-1}) \cap X_0| \leq n_r + n_b \leq m - q + 1.$$

Finally, for our contradiction, we show that for some s,

$$|\Gamma^-(W_s) \cap X_0| + |\Gamma^+(W_{s-1}) \cap X_0| \geq m - q + 2.$$

For every $W \in X_1$, since $d(W) \geq n$,

$$\begin{aligned} |\Gamma^-(W) \cap X_0| + |\Gamma^+(W) \cap X_0| &\geq |X_0| \\ &\quad - \Big(|\Gamma^-(W) \cap X_1| + |\Gamma^+(W) \cap X_1| - |X_1|\Big) \\ &\quad - \sum_{j \neq 0,1} \Big(|\Gamma^-(W) \cap X_j| + |\Gamma^+(W) \cap X_j| - |X_j|\Big) \\ &\geq m - ((q-1) + (q-1) - q) - 0 = m - q + 2. \end{aligned}$$

Now counting the arcs between X_0 and X_1 in two different ways, we have

$$\begin{aligned} \sum_{s=0}^{q-1} \Big(|\Gamma^-(W_s) \cap X_0| + |\Gamma^+(W_{s-1}) \cap X_0|\Big) &= \sum_{s=0}^{q-1} \Big(|\Gamma^-(W_s) \cap X_0| + |\Gamma^+(W_s) \cap X_0|\Big) \\ &\geq q(m - q + 2). \end{aligned}$$

Thus, for at least one s, the desired inequality holds. □

REFERENCES

[1] Claude Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1976.
[2] K. F. Fraughnaugh, J. R. Lundgren, S. K. Merz, J. S. Maybee, N. J. Pullman, *Competition graphs of strongly connected and Hamiltonian digraphs*, SIAM J. Discrete. Math., 8 (1995), pp. 179–185.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 135–156, February 1998

010

THE NUMBER OF INTERSECTION POINTS MADE BY THE DIAGONALS OF A REGULAR POLYGON*

BJORN POONEN† AND MICHAEL RUBINSTEIN‡

Abstract. We give a formula for the number of interior intersection points made by the diagonals of a regular n-gon. The answer is a polynomial on each residue class modulo 2520. We also compute the number of regions formed by the diagonals, by using Euler's formula $V - E + F = 2$.

Key words. regular polygons, diagonals, intersection points, roots of unity, adventitious quadrangles

AMS subject classifications. Primary, 51M04; Secondary, 11R18

PII. S0895480195281246

1. Introduction. We will find a formula for the number $I(n)$ of intersection points formed inside a regular n-gon by its diagonals. The case $n = 30$ is depicted in Figure 1.1. For a *generic* convex n-gon, the answer would be $\binom{n}{4}$, because every four vertices would be the endpoints of a unique pair of intersecting diagonals. But $I(n)$ can be less, because in a regular n-gon it may happen that three or more diagonals meet at an interior point, and then some of the $\binom{n}{4}$ intersection points will coincide. In fact, if n is even and at least 6, $I(n)$ will always be less than $\binom{n}{4}$ because there will be $n/2 \geq 3$ diagonals meeting at the center point. It will result from our analysis that for $n > 4$, the maximum number of diagonals of the regular n-gon that meet at a point other than the center is

$$\begin{array}{ll} 2 & \text{if } n \text{ is odd,} \\ 3 & \text{if } n \text{ is even but not divisible by 6,} \\ 5 & \text{if } n \text{ is divisible by 6 but not 30, and,} \\ 7 & \text{if } n \text{ is divisible by 30,} \end{array}$$

with two exceptions: this number is 2 if $n = 6$, and 4 if $n = 12$. In particular, it is impossible to have eight or more diagonals of a regular n-gon meeting at a point other than the center. Also, by our earlier remarks, the fact that no three diagonals meet when n is odd will imply that $I(n) = \binom{n}{4}$ for odd n.

A careful analysis of the possible configurations of three diagonals meeting will provide enough information to permit us in theory to deduce a formula for $I(n)$. But because the explicit description of these configurations is so complex, our strategy will be instead to use this information to deduce only the *form* of the answer, and then to compute the answer for enough small n that we can determine the result precisely. The computations are done in Mathematica, Maple, and C, and annotated source codes can be obtained at http://math.berkeley.edu/~poonen.

*Received by the editors February 8, 1995; accepted for publication (in revised form) November 22, 1996. Part of this research was completed at MSRI and was supported in part by NSF grant DMS-9022140.

http://www.siam.org/journals/sidma/11-1/28124.html

†AT&T Bell Laboratories, Murray Hill, NJ 07974. Current address: Department of Mathematics, University of California, Berkeley, CA 94720 (poonen@math.berkeley.edu). The research of this author was supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

‡AT&T Bell Laboratories, Murray Hill, NJ 07974. Current address: Department of Mathematics, Princeton University, Princeton, NJ 08544-1000 (miker@math.princeton.edu).

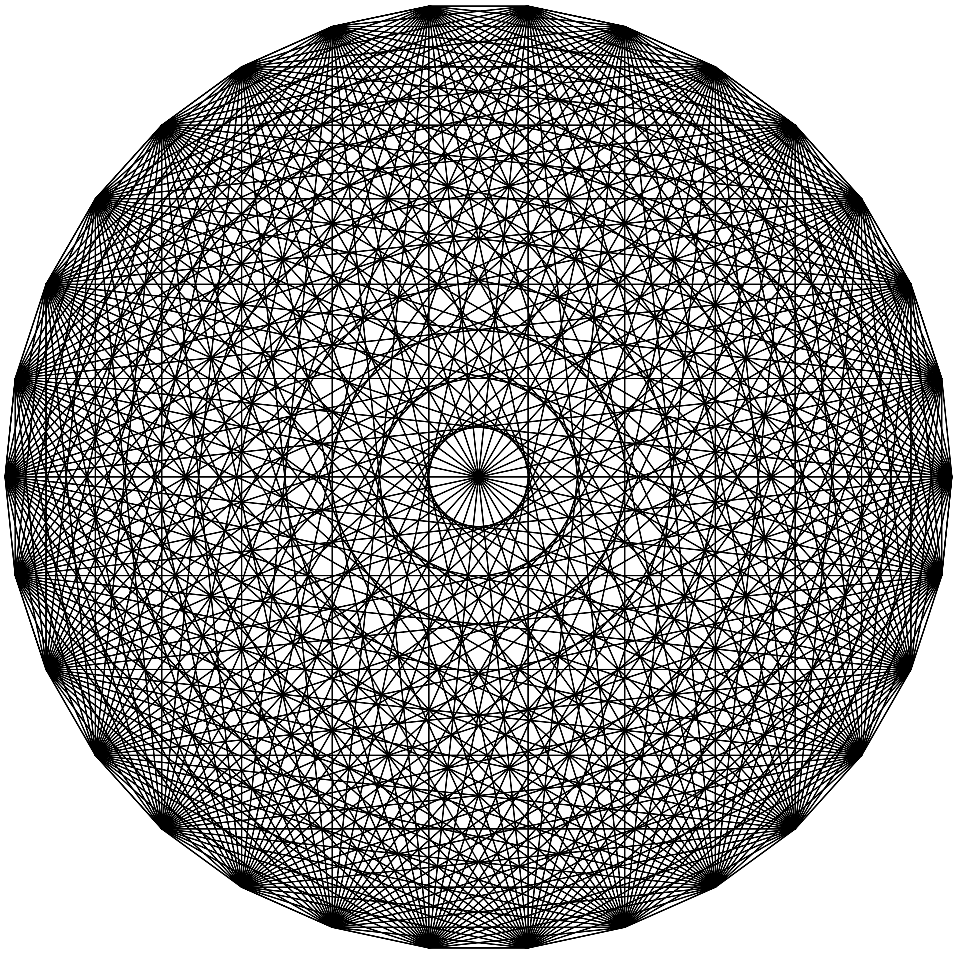


FIG. 1.1. *The* 30*-gon with its diagonals. There are* 16801 *interior intersection points:* 13800 *two line intersections,* 2250 *three line intersections,* 420 *four line intersections,* 180 *five line intersections,* 120 *six line intersections,* 30 *seven line intersections, and* 1 *fifteen line intersection.*

In order to write the answer in a reasonable form, we define

$$\delta_m(n) = \begin{cases} 1 & \text{if } n \equiv 0 \pmod{m}, \\ 0 & \text{otherwise.} \end{cases}$$

THEOREM 1.1. *For* $n \geq 3$,

$$\begin{aligned} I(n) = \binom{n}{4} &+ (-5n^3 + 45n^2 - 70n + 24)/24 \cdot \delta_2(n) - (3n/2) \cdot \delta_4(n) \\ &+ (-45n^2 + 232n)/6 \cdot \delta_6(n) + 42n \cdot \delta_{12}(n) + 60n \cdot \delta_{18}(n) \\ &+ 35n \cdot \delta_{24}(n) - 38n \cdot \delta_{30}(n) - 82n \cdot \delta_{42}(n) - 330n \cdot \delta_{60}(n) \\ &- 144n \cdot \delta_{84}(n) - 96n \cdot \delta_{90}(n) - 144n \cdot \delta_{120}(n) - 96n \cdot \delta_{210}(n). \end{aligned}$$

Further analysis, involving Euler's formula $V - E + F = 2$, will yield a formula for the number $R(n)$ of regions that the diagonals cut the n-gon into.

THEOREM 1.2. *For $n \geq 3$,*

$$\begin{aligned} R(n) = {} & (n^4 - 6n^3 + 23n^2 - 42n + 24)/24 \\ & + (-5n^3 + 42n^2 - 40n - 48)/48 \cdot \delta_2(n) - (3n/4) \cdot \delta_4(n) \\ & + (-53n^2 + 310n)/12 \cdot \delta_6(n) + (49n/2) \cdot \delta_{12}(n) + 32n \cdot \delta_{18}(n) \\ & + 19n \cdot \delta_{24}(n) - 36n \cdot \delta_{30}(n) - 50n \cdot \delta_{42}(n) - 190n \cdot \delta_{60}(n) \\ & - 78n \cdot \delta_{84}(n) - 48n \cdot \delta_{90}(n) - 78n \cdot \delta_{120}(n) - 48n \cdot \delta_{210}(n). \end{aligned}$$

These problems have been studied by many authors before, but this is apparently the first time the correct formulas have been obtained. The Dutch mathematician Gerrit Bol [1] gave a complete solution in 1936, except that a few of the coefficients in his formulas are wrong. (A few misprints and omissions in Bol's paper are mentioned in [11].)

The approaches used by us and Bol are similar in many ways. One difference (which is not too substantial) is that we work as much as possible with roots of unity, whereas Bol tended to use more trigonometry (integer relations between sines of rational multiples of π). Also, we relegate much of the work to the computer, whereas Bol had to enumerate the many cases by hand. The task is so formidable that it is amazing to us that Bol was able to complete it, and at the same time not so surprising that it would contain a few errors!

Bol's work was largely forgotten. In fact, even we were not aware of his paper until after deriving the formulas ourselves. Many other authors in the interim solved special cases of the problem. Steinhaus [14] posed the problem of showing that no three diagonals meet internally when n is prime, and this was solved by Croft and Fowler [3]. (Steinhaus also mentions this in [13], which includes a picture of the 23-gon and its diagonals.) In the 1960s, Heineken [6] gave a delightful argument which generalized this to all odd n, and later he [7] and Harborth [4] independently enumerated all three-diagonal intersections for n not divisible by 6.

The classification of three-diagonal intersections also solves Colin Tripp's problem [15] of enumerating "adventitious quadrilaterals," those convex quadrilaterals for which the angles formed by sides and diagonals are all rational multiples of π. See Rigby's paper [11] or the summary [10] for details. Rigby, who was aware of Bol's work, mentions that Monsky and Pleasants also each independently classified all three-diagonal intersections of regular n-gons. Rigby's papers partially solve Tripp's further problem of proving the existence of all adventitious quadrangles using only elementary geometry; i.e., without resorting to trigonometry.

All the questions so far have been in the Euclidean plane. What happens if we count the interior intersections made by the diagonals of a hyperbolic regular n-gon? The answers are exactly the same, as pointed out in [11], because if we use Beltrami's representation of points of the hyperbolic plane by points inside a circle in the Euclidean plane, we can assume that the center of the hyperbolic n-gon corresponds to the center of the circle, and then the hyperbolic n-gon with its diagonals looks in the model exactly like a Euclidean regular n-gon with its diagonals. It is equally easy to see that the answers will be the same in elliptic geometry.

2. When do three diagonals meet? We now begin our derivations of the formulas for $I(n)$ and $R(n)$. The first step will be to find a criterion for the concurrency of three diagonals. Let A, B, C, D, E, F be six distinct points in order on a unit circle

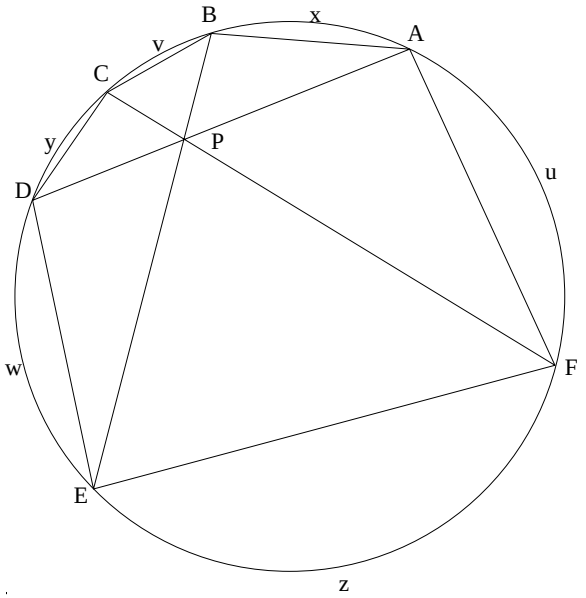


FIG. 2.1.

dividing up the circumference into arc lengths u, x, v, y, w, z and assume that the three chords AD, BE, CF meet at P (see Figure 2.1).

By similar triangles, $AF/CD = PF/PD$, $BC/EF = PB/PF$, $DE/AB = PD/PB$. Multiplying these together yields

$$(AF \cdot BC \cdot DE)/(CD \cdot EF \cdot AB) = 1,$$

and so

$$\sin(u/2)\sin(v/2)\sin(w/2) = \sin(x/2)\sin(y/2)\sin(z/2). \tag{2.1}$$

Conversely, suppose six distinct points A, B, C, D, E, F partition the circumference of a unit circle into arc lengths u, x, v, y, w, z and suppose that (2.1) holds. Then the three diagonals AD, BE, CF meet in a single point which we see as follows. Let lines AD and BE intersect at P_0. Form the line through F and P_0 and let C' be the other intersection point of FP_0 with the circle. This partitions the circumference into arc lengths u, x, v', y', w, z. As shown above, we have

$$\sin(u/2)\sin(v'/2)\sin(w/2) = \sin(x/2)\sin(y'/2)\sin(z/2),$$

and since we are assuming that (2.1) holds for u, x, v, y, w, z we get

$$\frac{\sin(v'/2)}{\sin(y'/2)} = \frac{\sin(v/2)}{\sin(y/2)}.$$

Let $\alpha = v + y = v' + y'$. Substituting $v = \alpha - y$, $v' = \alpha - y'$ above we get

$$\frac{\sin(\alpha/2)\cos(y'/2) - \cos(\alpha/2)\sin(y'/2)}{\sin(y'/2)} = \frac{\sin(\alpha/2)\cos(y/2) - \cos(\alpha/2)\sin(y/2)}{\sin(y/2)},$$

and so

$$\cot(y'/2) = \cot(y/2).$$

Now $0 < \alpha/2 < \pi$, so $y = y'$ and hence $C = C'$. Thus, the three diagonals AD, BE, CF meet at a single point.

So (2.1) gives a necessary and sufficient condition (in terms of arc lengths) for the chords AD, BE, CF formed by six distinct points A, B, C, D, E, F on a unit circle to meet at a single point. In other words, to give an explicit answer to the question in the section title, we need to characterize the positive rational solutions to

$$\begin{aligned} &\sin(\pi U)\sin(\pi V)\sin(\pi W) = \sin(\pi X)\sin(\pi Y)\sin(\pi Z) \\ &U + V + W + X + Y + Z = 1. \end{aligned} \tag{2.2}$$

(Here $U = u/(2\pi)$, etc.) This is a trigonometric diophantine equation in the sense of [2], where it is shown that in theory, there is a finite computation which reduces the solution of such equations to ordinary diophantine equations. The solutions to the analogous equation with only two sines on each side are listed in [9].

If in (2.2) we substitute $\sin(\theta) = (e^{i\theta} - e^{-i\theta})/(2i)$, multiply both sides by $(2i)^3$, and expand, we get a sum of eight terms on the left equaling a similar sum on the right, but two terms on the left cancel with two terms on the right since $U+V+W = 1-(X+Y+Z)$, leaving

$$\begin{aligned} &-e^{i\pi(V+W-U)} + e^{-i\pi(V+W-U)} - e^{i\pi(W+U-V)} \\ &+e^{-i\pi(W+U-V)} - e^{i\pi(U+V-W)} + e^{-i\pi(U+V-W)} \\ =\ &-e^{i\pi(Y+Z-X)} + e^{-i\pi(Y+Z-X)} - e^{i\pi(Z+X-Y)} \\ &+e^{-i\pi(Z+X-Y)} - e^{i\pi(X+Y-Z)} + e^{-i\pi(X+Y-Z)}. \end{aligned}$$

If we move all terms to the left-hand side, convert minus signs into $e^{-i\pi}$, multiply by $i = e^{i\pi/2}$, and let

$$\begin{aligned} \alpha_1 &= V + W - U - 1/2, \\ \alpha_2 &= W + U - V - 1/2, \\ \alpha_3 &= U + V - W - 1/2, \\ \alpha_4 &= Y + Z - X + 1/2, \\ \alpha_5 &= Z + X - Y + 1/2, \\ \alpha_6 &= X + Y - Z + 1/2, \end{aligned}$$

we obtain

$$\sum_{j=1}^{6} e^{i\pi\alpha_j} + \sum_{j=1}^{6} e^{-i\pi\alpha_j} = 0, \tag{2.3}$$

in which $\sum_{j=1}^{6} \alpha_j = U+V+W+X+Y+Z = 1$. Conversely, given rational numbers $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ (not necessarily positive) which sum to 1 and satisfy (2.3), we can recover U, V, W, X, Y, Z, (for example, $U = (\alpha_2+\alpha_3)/2+1/2$), but we must check that they turn out positive.

3. Zero as a sum of 12 roots of unity. In order to enumerate the solutions to (2.2), we are led, as in the end of the last section, to classify the ways in which 12 roots of unity can sum to zero. More generally, we will study relations of the form

$$\sum_{i=1}^{k} a_i \eta_i = 0, \tag{3.1}$$

where the a_i are positive integers, and the η_i are distinct roots of unity. (These have been studied previously by Schoenberg [12], Mann [8], Conway and Jones [2], and others.) We call $w(S) = \sum_{i=1}^{k} a_i$ the *weight* of the relation S. (So we shall be particularly interested in relations of weight 12.) We shall say the relation (3.1) is *minimal* if it has no nontrivial subrelation; i.e., if

$$\sum_{i=1}^{k} b_i \eta_i = 0, \quad a_i \geq b_i \geq 0$$

implies either $b_i = a_i$ for all i or $b_i = 0$ for all i. By induction on the weight, any relation can be represented as a sum of minimal relations (but the representation need not be unique).

Let us give some examples of minimal relations. For each $n \geq 1$, let $\zeta_n = \exp(2\pi i/n)$ be the standard primitive nth root of unity. For each prime p, let R_p be the relation

$$1 + \zeta_p + \zeta_p^2 + \cdots + \zeta_p^{p-1} = 0.$$

Its minimality follows from the irreducibility of the cyclotomic polynomial. Also, we can "rotate" any relation by multiplying through by an arbitrary root of unity to obtain a new relation. In fact, Schoenberg [12] proved that every relation (even those with possibly negative coefficients) can be obtained as a linear combination with positive and negative integral coefficients of the R_p and their rotations. But we are only allowing positive combinations, so it is not clear that these are enough to generate all relations.

In fact it is not even true! In other words, there are other minimal relations. If we subtract R_3 from R_5, cancel the 1's and incorporate the minus signs into the roots of unity, we obtain a new relation

$$\zeta_6 + \zeta_6^{-1} + \zeta_5 + \zeta_5^2 + \zeta_5^3 + \zeta_5^4 = 0, \tag{3.2}$$

which we will denote $(R_5 : R_3)$. In general, if S and $T_1, T_2, \ldots, T_j$ are relations, we will use the notation $(S : T_1, T_2, \ldots, T_j)$ to denote any relation obtained by rotating the T_i so that each shares exactly one root of unity with S which is different for each i, subtracting them from S, and incorporating the minus signs into the roots of unity. For notational convenience, we will write $(R_5 : 4R_3)$ for $(R_5 : R_3, R_3, R_3, R_3)$, for example. Note that although $(R_5 : R_3)$ denotes unambiguously (up to rotation) the relation listed in (3.2), in general there will be many relations of type $(S : T_1, T_2, \ldots, T_j)$ up to rotational equivalence. Let us also remark that including R_2's in the list of T's has no effect.

It turns out that recursive use of the construction above is enough to generate all minimal relations of weight up to 12. These are listed in Table 3.1. The completeness and correctness of the table will be proved in Theorem 3.1 below. Although there are

TABLE 3.1
The 107 *minimal relations of weight up to* 12.

Weight	Relation type	Number of relations of that type
2	R_2	1
3	R_3	1
5	R_5	1
6	$(R_5 : R_3)$	1
7	$(R_5 : 2R_3)$	2
	R_7	1
8	$(R_5 : 3R_3)$	2
	$(R_7 : R_3)$	1
9	$(R_5 : 4R_3)$	1
	$(R_7 : 2R_3)$	3
10	$(R_7 : 3R_3)$	5
	$(R_7 : R_5)$	1
11	$(R_7 : 4R_3)$	5
	$(R_7 : R_5, R_3)$	6
	$(R_7 : (R_5 : R_3))$	6
	R_{11}	1
12	$(R_7 : 5R_3)$	3
	$(R_7 : R_5, 2R_3)$	15
	$(R_7 : (R_5 : R_3), R_3)$	36
	$(R_7 : (R_5 : 2R_3))$	14
	$(R_{11} : R_3)$	1

107 minimal relations up to rotational equivalence, often the minimal relations within one of our classes are Galois conjugates. For example, the two minimal relations of type $(R_5 : 2R_3)$ are conjugate under $\mathrm{Gal}(\mathbb{Q}(\zeta_{15})/\mathbb{Q})$, as pointed out in [8].

The minimal relations with $k \leq 7$ (k defined as in (3.1)) had been previously catalogued in [8], and those with $k \leq 9$ in [2]. In fact, the a_i in these never exceed 1, so these also have weight less than or equal to 9.

THEOREM 3.1. *Table* 3.1 *is a complete listing of the minimal relations of weight up to* 12 *(up to rotation).*

The following three lemmas will be needed in the proof.

LEMMA 3.2. *If the relation* (3.1) *is minimal, then there are distinct primes* $p_1 < p_2 < \cdots < p_s \leq k$ *so that each* η_i *is a* $p_1 p_2 \cdots p_s$*-th root of unity, after the relation has been suitably rotated.*

Proof. This is a corollary of Theorem 1 in [8]. □

LEMMA 3.3. *The only minimal relations (up to rotation) involving only the* $2p$*-th roots of unity, for* p *prime, are* R_2 *and* R_p*.*

Proof. Any $2p$-th root of unity is of the form $\pm\zeta^i$. If both $+\zeta^i$ and $-\zeta^i$ occurred in the same relation, then R_2 occurs as a subrelation. So the relation has the form

$$\sum_{i=0}^{p-1} c_i \zeta_p^i = 0.$$

By the irreducibility of the cyclotomic polynomial, $\{1, \zeta_p, \ldots, \zeta_p^{p-1}\}$ are independent over $\mathbb{Q}$ save for the relation that their sum is zero, so all the c_i must be equal. If they are all positive, then R_p occurs as a subrelation. If they are all negative, then R_p rotated by -1 (i.e., 180 degrees) occurs as a subrelation. □

LEMMA 3.4. *Suppose* S *is a minimal relation, and* $p_1 < p_2 < \cdots < p_s$ *are picked as in Lemma* 3.2 *with* $p_1 = 2$ *and* p_s *minimal. If* $w(S) < 2p_s$*, then* S *(or a rotation)*

is of the form $(R_{p_s} : T_1, T_2, \ldots, T_j)$ *where the* T_i *are minimal relations not equal to* R_2 *and involving only* $p_1p_2\cdots p_{s-1}$*-th roots of unity, such that* $j < p_s$ *and*

$$\sum_{i=1}^{j}[w(T_i) - 2] = w(S) - p_s.$$

Proof. Since every $p_1p_2\cdots p_s$-th root of unity is uniquely expressible as the product of a $p_1p_2\cdots p_{s-1}$-th root of unity and a p_s-th root of unity, the relation can be rewritten as

$$\sum_{i=0}^{p_s-1} f_i\zeta_{p_s}^i = 0, \tag{3.3}$$

where each f_i is a sum of $p_1p_2\cdots p_{s-1}$-th roots of unity, which we will think of as a sum (not just its value).

Let K_m be the field obtained by adjoining the $p_1p_2\cdots p_m$-th roots of unity to $\mathbb{Q}$. Since $[K_s : K_{s-1}] = \phi(p_1p_2\cdots p_s)/\phi(p_1p_2\cdots p_{s-1}) = \phi(p_s) = p_s - 1$, the only linear relation satisfied by $1, \zeta_{p_s}, \ldots, \zeta_{p_s}^{p_s-1}$ over K_{s-1} is that their sum is zero. Hence (3.3) forces the values of the f_i to be equal.

The total number of roots of unity in all the f_i's is $w(S) < 2p_s$, so by the pigeonhole principle, some f_i is zero or consists of a single root of unity. In the former case, each f_j sums to zero, but at least two of these sums contain at least one root of unity, since otherwise s was not minimal, so one of these sums gives a subrelation of S, contradicting its minimality. So some f_i consists of a single root of unity. By rotation, we may assume $f_0 = 1$. Then each f_i sums to 1, and if it is not simply the single root of unity 1, the negatives of the roots of unity in f_i together with 1 form a relation T_i which is not R_2 and involves only $p_1p_2\cdots p_{s-1}$-th roots of unity, and it is clear that S is of type $(R_{p_s} : T_{i_1}, T_{i_2}, \ldots, T_{i_j})$. If one of the T's were not minimal, then it could be decomposed into two nontrivial subrelations, one of which would not share a root of unity with the R_{p_s}, and this would give a nontrivial subrelation of S, contradicting the minimality of S. Finally, $w(S)$ must equal the sum of the weights of R_{p_s} and the T's, minus $2j$ to account for the roots of unity that are cancelled in the construction of $(R_{p_s} : T_{i_1}, T_{i_2}, \ldots, T_{i_j})$. □

Proof of Theorem 3.1. We will content ourselves with proving that every relation of weight up to 12 can be decomposed into a sum of the ones listed in Table 3.1, it then being straightforward to check that the entries in the table are distinct and that none of them can be further decomposed into relations higher up in the table.

Let S be a minimal relation with $w(S) \le 12$. Pick $p_1 < p_2 < \cdots < p_s$ as in Lemma 3.2 with $p_1 = 2$ and p_s minimal. In particular, $p_s \le 12$, so $p_s = 2,3,5,7,$ or 11.

Case 1. $p_s \le 3$.

Here the only minimal relations are R_2 and R_3, by Lemma 3.3.

Case 2. $p_s = 5$.

If $w(S) < 10$, then we may apply Lemma 3.4 to deduce that S is of type $(R_5 : T_1, T_2, \ldots, T_j)$. Each T must be R_3 (since $p_{s-1} \le 3$), and $j = w(S) - 5$ by the last equation in Lemma 3.4. The number of relations of type $(R_5 : jR_3)$, up to rotation, is $\binom{5}{j}/5$. (There are $\binom{5}{j}$ ways to place the R_3's, but one must divide by 5 to avoid counting rotations of the same relation.)

If $10 \le w(S) \le 12$, then write S as in (3.3). If some f_i consists of zero or one roots of unity, then the argument of Lemma 3.4 applies, and S must be of the form

$(R_5 : jR_3)$ with $j \le 4$, which contradicts the last equation in the lemma. Otherwise, the numbers of (sixth) roots of unity occurring in f_0, f_1, f_2, f_3, f_4 must be 2,2,2,2,2 or 2,2,2,2,3 or 2,2,2,3,3 or 2,2,2,2,4 in some order. So the common value of the f_i is a sum of two sixth roots of unity. By rotating by a sixth root of unity, we may assume this value is 0, 1, $1 + \zeta_6$, or 2. If it is 0 or 1, then the arguments in the proof of Lemma 3.4 apply. Next assume it is $1+\zeta_6$. The only way two sixth roots of unity can sum to $1 + \zeta_6$ is if they are 1 and ζ_6 in some order. The only way three sixth roots of unity can sum to $1 + \zeta_6$ is if they are $1, 1, \zeta_6^2$ or $\zeta_6, \zeta_6, \zeta_6^{-1}$. So if the numbers of roots of unity occurring in f_0, f_1, f_2, f_3, f_4 are 2,2,2,2,2 or 2,2,2,2,3, then S will contain R_5 or its rotation by ζ_6, and the same will be true for 2,2,2,3,3 unless the two f_i with three terms are $1 + 1 + \zeta_6^2$ and $\zeta_6 + \zeta_6 + \zeta_6^{-1}$, in which case S contains $(R_5 : R_3)$. It is impossible to write $1 + \zeta_6$ as a sum of sixth roots of unity without using 1 or ζ_6, so if the numbers are 2,2,2,2,4, then again S contains R_5 or its rotation by ζ_6. Thus we get no new relations where the common value of the f_i is $1 + \zeta_6$. Lastly, assume this common value is 2. Any representation of 2 as a sum of four or fewer sixth roots of unity contains 1, unless it is $\zeta_6 + \zeta_6 + \zeta_6^{-1} + \zeta_6^{-1}$, so S will contain R_5 except possibly in the case where f_0, f_1, f_2, f_3, f_4 are 2,2,2,2,4 in some order, and the 4 is as above. But in this final remaining case, S contains $(R_5 : R_3)$. Thus there are no minimal relations S with $p_s = 5$ and $10 \le w(S) \le 12$.

Case 3. $p_s = 7$.

Since $w(S) \le 12 < 2 \cdot 7$, we can apply Lemma 3.4. Now the sum of $w(T_i) - 2$ is required to be $w(S) - 7$ which is at most 5, so the T's that may be used are R_3, R_5, $(R_5 : R_3)$, and the two of type $(R_5 : 2R_3)$, for which weight minus 2 equals 1, 3, 4, and 5, respectively. So the problem is reduced to listing the partitions of $w(S) - 7$ into parts of size 1, 3, 4, and 5.

If all parts used are 1, then we get $(R_7 : jR_3)$ with $j = w(S) - 7$, and there are $\binom{7}{j}/7$ distinct relations in this class. Otherwise exactly one part of size 3, 4, or 5 is used, and the possibilities are as follows. If a part of size 3 is used, we get $(R_7 : R_5)$, $(R_7 : R_5, R_3)$, or $(R_7 : R_5, 2R_3)$, of weights 10, 11, and 12, respectively. By rotation, the R_5 may be assumed to share the 1 in the R_7, and then there are $\binom{6}{i}$ ways to place the R_3's where i is the number of R_3's. If a part of size 4 is used, we get $(R_7 : (R_5 : R_3))$ of weight 11 or $(R_7 : (R_5 : R_3), R_3)$ of weight 12. By rotation, the $(R_5 : R_3)$ may be assumed to share the 1 in the R_7, but any of the six roots of unity in the $(R_5 : R_3)$ may be rotated to be 1. The R_3 can then overlap any of the other six seventh roots of unity. Finally, if a part of size 5 is used, we get $(R_7 : (R_5 : 2R_3))$. There are two different relations of type $(R_5 : 2R_3)$ that may be used, and each has seven roots of unity which may be rotated to be the 1 shared by the R_7, so there are 14 of these all together.

Case 4. $p_s = 11$.

Applying Lemma 3.4 shows that the only possibilities are R_{11} of weight 11 and $(R_{11} : R_3)$ of weight 12. □

Now a general relation of weight 12 is a sum of the minimal ones of weight up to 12, and we can classify them according to the weights of the minimal relations, which form a partition of 12 with no parts of size 1 or 4. We will use the notation $(R_5 : 2R_3)+2R_3$, for example, to denote a sum of three minimal relations of type $(R_5 : 2R_3)$, R_3, and R_3. Table 3.2 lists the possibilities. The parts may be rotated independently, so any category involving more than one minimal relation contains infinitely many relations, even up to rotation (of the entire relation). Also, the categories are not mutually exclusive because of the nonuniqueness of the decomposition into minimal relations.

TABLE 3.2
The types of relations of weight 12.

Partition	Relation type
12	$(R_7 : 5R_3)$
	$(R_7 : R_5, 2R_3)$
	$(R_7 : (R_5 : R_3), R_3)$
	$(R_7 : (R_5 : 2R_3))$
	$(R_{11} : R_3)$
10,2	$(R_7 : 3R_3) + R_2$
	$(R_7 : R_5) + R_2$
9,3	$(R_5 : 4R_3) + R_3$
	$(R_7 : 2R_3) + R_3$
8,2,2	$(R_5 : 3R_3) + 2R_2$
	$(R_7 : R_3) + 2R_2$

Partition	Relation type
7,5	$(R_5 : 2R_3) + R_5$
	$R_7 + R_5$
7,3,2	$(R_5 : 2R_3) + R_3 + R_2$
	$R_7 + R_3 + R_2$
6,6	$2(R_5 : R_3)$
6,3,3	$(R_5 : R_3) + 2R_3$
6,2,2,2	$(R_5 : R_3) + 3R_2$
5,5,2	$2R_5 + R_2$
5,3,2,2	$R_5 + R_3 + 2R_2$
3,3,3,3	$4R_3$
3,3,2,2,2	$2R_3 + 3R_2$
2,2,2,2,2,2	$6R_2$

4. Solutions to the trigonometric equation. Here we use the classification of the previous section to give a complete listing of the solutions to the trigonometric equation (2.2). There are some obvious solutions to (2.2), namely those in which U, V, W are arbitrary positive rational numbers with sum $1/2$, and X, Y, Z are a permutation of U, V, W. We will call these the trivial solutions, even though the three-diagonal intersections they give rise to can look surprising. See Figure 4.1 for an example on the 16-gon.

The twelve roots of unity occurring in (2.3) are not arbitrary; therefore we must go through Table 3.2 to see which relations are of the correct form, i.e., expressible as a sum of six roots of unity and their inverses, where the product of the six is -1. First let us prove a few lemmas that will greatly reduce the number of cases.

LEMMA 4.1. *Let S be a relation of weight $k \le 12$. Suppose S is stable under complex conjugation (i.e., under $\zeta \mapsto \zeta^{-1}$). Then S has a complex conjugation-stable decomposition into minimal relations; i.e., each minimal relation occurring is itself stable under complex conjugation or can be paired with another minimal relation which is its complex conjugate.*

Proof. We will use induction on k. If S is minimal, there is nothing to prove. Otherwise let T be a (minimal) subrelation of S of minimal weight, so T is of weight at most 6. The complex conjugate $\overline{T}$ of T is another minimal relation in S. If they do not intersect, then we take the decomposition of S into T, $\overline{T}$, and a decomposition of $S \setminus (T \cup \overline{T})$ given by the inductive hypothesis. If they do overlap and the weight of T is at most 5, then $T = R_p$ for some prime p, and the fact that T intersects $\overline{T}$ implies that $T = \overline{T}$, and we get the result by applying the inductive hypothesis to $S \setminus T$.

The only remaining case is where S is of type $2(R_5 : R_3)$. If the two $(R_5 : R_3)$'s are not conjugate to each other, then for each there is a root of unity ζ such that ζ and ζ^{-1} occur in that (rotation of) $(R_5 : R_3)$. The quotient ζ^2 is then a 30th root of unity, so ζ itself is a 60th root of unity. Thus each $(R_5 : R_3)$ is a rotation of the "standard" $(R_5 : R_3)$ as in (3.2) by a 60th root of unity, and we let Mathematica check the 60^2 possibilities. □

We do not know if the preceding lemma holds for relations of weight greater than 12.

LEMMA 4.2. *Let S be a minimal relation of type $(R_p : T_1, \ldots, T_j)$, $p \ge 5$, where the T_i involve roots of unity of order prime to p, and $j < p$. If S is stable under complex conjugation, then the particular rotation of R_p from which the T_i were "subtracted" is also stable (and hence so is the collection of the relations subtracted).*

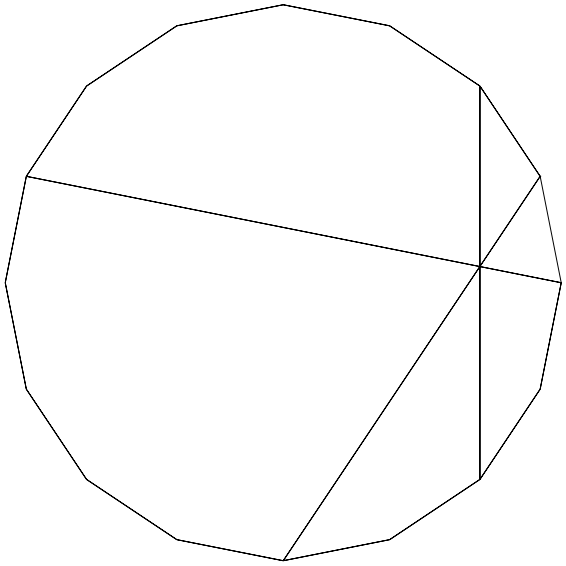


FIG. 4.1. *A surprising trivial solution for the* 16*-gon. The intersection point does not lie on any of the* 16 *lines of symmetry of the* 16*-gon.*

Proof. Let ℓ be the product of the orders of the roots of unity in all the T_i. The elements of S in the original R_p can be characterized as those terms of S that are unique in their coset of μ_ℓ (the ℓth roots of unity), and this condition is stable under complex conjugation, so the set of terms of the R_p that were not subtracted is stable. Since $j < p$, we can pick one such term ζ. Then the quotient ζ/ζ^{-1} is a pth root of unity, so ζ is a $2p$-th root of unity, and hence the R_p containing it is stable. ∎

COROLLARY 4.3. *A relation of type* $(R_7 : (R_5 : R_3), R_3)$ *cannot be stable under complex conjugation.*

Even with these restrictions, a very large number of cases remain, so we perform the calculation using Mathematica. Each entry of Table 3.2 represents a finite number of linearly parameterized (in the exponents) families of relations of weight 12. For each parameterized family, we check to see what additional constraints must be put on the parameters for the relation to be of the form of (2.3). Next, for each parameterized family of solutions to (2.3), we calculate the corresponding U, V, W, X, Y, Z and throw away solutions in which some of these are nonpositive. Finally, we sort U, V, W and X, Y, Z and interchange the two triples if $U > X$ in order to count the solutions only up to symmetry.

The results of this computation are recorded in the following theorem.

THEOREM 4.4. *The positive rational solutions to* (2.2)*, up to symmetry, can be classified as follows:*

1. *The trivial solutions, which arise from relations of type* $6R_2$.
2. *Four one-parameter families of solutions, listed in Table* 4.1*. The first arises from relations of type* $4R_3$*, and the other three arise from relations of type* $2R_3 + 3R_2$.
3. *Sixty-five "sporadic" solutions, listed in Table* 4.2*, which arise from the other types of weight* 12 *relations listed in Table* 3.2.

The only duplications in this list are that the second family of Table 4.1 *gives a trivial*

TABLE 4.1
The nontrivial infinite families of solutions to (2.2).

U	V	W	X	Y	Z	Range
$1/6$	t	$1/3-2t$	$1/3+t$	t	$1/6-t$	$0<t<1/6$
$1/6$	$1/2-3t$	t	$1/6-t$	$2t$	$1/6+t$	$0<t<1/6$
$1/6$	$1/6-2t$	$2t$	$1/6-2t$	t	$1/2+t$	$0<t<1/12$
$1/3-4t$	t	$1/3+t$	$1/6-2t$	$3t$	$1/6+t$	$0<t<1/12$

solution for $t=1/12$, the first and fourth families of Table 4.1 *give the same solution when $t=1/18$ in both, and the second and fourth families of Table* 4.1 *give the same solution when $t=1/24$ in both.*

Some explanation of the tables is in order. The last column of Table 4.1 gives the allowable range for the rational parameter t. The entries of Table 4.2 are sorted according to the least common denominator of U, V, W, X, Y, Z, which is also the least n for which diagonals of a regular n-gon can create arcs of the corresponding lengths. The relation type from which each solution derives is also given. The reason 11 does not appear in the least common denominator for any sporadic solution is that the relation $(R_{11} : R_3)$ cannot be put in the form of (2.3) with the α_j summing to 1, and hence leads to no solutions of (2.2). (Several other types of relations also give rise to no solutions.)

Tables 4.1 and 4.2 are the same as Bol's tables at the bottom of page 40 and on page 41 of [1], in a slightly different format.

The arcs cut by diagonals of a regular n-gon have lengths which are multiples of $2\pi/n$, so U, V, W, X, Y and Z corresponding to any configuration of three diagonals meeting must be multiples of $1/n$. With this additional restriction, trivial solutions to (2.2) occur only when n is even (and at least 6). Solutions within the infinite families of Table 4.1 occur when n is a multiple of 6 (and at least 12), and there t must be a multiple of $1/n$. Sporadic solutions with least common denominator d occur if and only if n is a multiple of d.

5. Intersections of more than three diagonals. Now that we know the configurations of three diagonals meeting, we can check how they overlap to produce configurations of more than three diagonals meeting. We will disregard configurations in which the intersection point is the center of the n-gon, since these are easily described: there are exactly $n/2$ diagonals (diameters) through the center when n is even, and none otherwise.

When k diagonals meet, they form $2k$ arcs, whose lengths we will measure as a fraction of the whole circumference (so they will be multiples of $1/n$) and list in counterclockwise order. (Warning: this is different from the order used in Tables 4.1 and 4.2.) The least common denominator of the numbers in this list will be called the denominator of the configuration. It is the least n for which the configuration can be realized as diagonals of a regular n-gon.

LEMMA 5.1. *If a configuration of $k \geq 2$ diagonals meeting at an interior point other than the center has denominator dividing d, then any configuration of diagonals meeting at that point has denominator dividing* $\mathrm{LCM}(2d, 3)$.

Proof. We may assume $k = 2$. Any other configuration of diagonals through the intersection point is contained in the union of configurations obtained by adding one diagonal to the original two, so we may assume the final configuration consists of three diagonals, two of which were the original two. Now we need only go through our list of three-diagonal intersections.

TABLE 4.2
The 65 *sporadic solutions to* (2.2).

Denominator	U	V	W	X	Y	Z	Relation type
30	1/10	2/15	3/10	2/15	1/6	1/6	$2(R_5 : R_3)$
	1/15	1/15	7/15	1/15	1/10	7/30	
	1/30	7/30	4/15	1/15	1/10	3/10	
	1/30	1/10	7/15	1/15	1/15	4/15	
	1/30	1/15	19/30	1/15	1/10	1/10	
	1/15	1/6	4/15	1/10	1/10	3/10	$(R_5 : R_3) + 2R_3$
	1/15	2/15	11/30	1/10	1/6	1/6	
	1/30	1/6	13/30	1/10	2/15	2/15	
	1/30	1/30	7/10	1/30	1/15	2/15	
	1/30	7/30	3/10	1/15	2/15	7/30	$R_5 + R_3 + 2R_2$
	1/30	1/6	11/30	1/15	1/10	4/15	
	1/30	1/10	13/30	1/30	2/15	4/15	
	1/30	1/15	8/15	1/30	1/10	7/30	
42	1/14	5/42	5/14	2/21	5/42	5/21	$(R_7 : 5R_3)$
	1/21	4/21	13/42	1/14	1/6	3/14	
	1/42	3/14	5/14	1/21	1/6	4/21	
	1/42	1/6	19/42	1/14	2/21	4/21	
	1/42	1/6	13/42	1/21	1/14	8/21	
	1/42	1/21	13/21	1/42	1/14	3/14	
60	1/20	1/12	29/60	1/15	1/10	13/60	$2(R_5 : R_3)$
	1/20	1/12	9/20	1/15	1/12	4/15	
	1/20	1/12	5/12	1/20	1/10	3/10	
	1/60	4/15	3/10	1/20	1/12	17/60	
	1/60	13/60	9/20	1/12	1/10	2/15	
	1/60	13/60	5/12	1/20	2/15	1/6	
	1/12	1/6	17/60	2/15	3/20	11/60	$(R_5 : 3R_3) + 2R_2$
	1/12	2/15	19/60	1/10	3/20	13/60	
	1/15	11/60	13/60	1/12	1/10	7/20	
	1/20	11/60	3/10	1/12	7/60	4/15	
	1/20	1/10	23/60	1/15	1/12	19/60	
	1/30	7/60	19/60	1/20	1/15	5/12	
	1/30	1/12	7/12	1/15	1/10	2/15	
	1/30	1/20	11/20	1/30	1/15	4/15	
	1/60	3/10	7/20	1/12	7/60	2/15	
	1/60	4/15	23/60	1/12	1/10	3/20	
	1/60	7/30	5/12	1/15	7/60	3/20	
	1/60	13/60	11/30	1/20	1/12	4/15	
	1/60	1/6	31/60	1/15	1/10	2/15	
	1/60	1/6	5/12	1/20	1/15	17/60	
	1/60	2/15	9/20	1/30	1/12	17/60	
	1/60	1/10	31/60	1/30	1/15	4/15	
84	1/12	3/14	19/84	11/84	13/84	4/21	$(R_7 : R_3) + 2R_2$
	1/14	11/84	23/84	1/12	2/21	29/84	
	1/21	13/84	23/84	1/14	1/12	31/84	
	1/42	1/12	7/12	1/21	1/14	4/21	
	1/84	25/84	5/14	5/84	1/12	4/21	
	1/84	5/21	5/12	5/84	1/14	17/84	
	1/84	3/14	37/84	1/21	1/12	17/84	
	1/84	1/6	43/84	1/21	1/14	4/21	
90	1/18	13/90	7/18	11/90	2/15	7/45	$(R_5 : R_3) + 2R_3$
	1/45	19/90	16/45	1/18	1/10	23/90	
	1/90	23/90	31/90	2/45	1/15	5/18	
	1/90	17/90	47/90	1/18	4/45	2/15	
120	13/120	3/20	31/120	2/15	19/120	23/120	$(R_5 : R_3) + 3R_2$
	1/12	19/120	29/120	1/10	13/120	37/120	
	1/20	23/120	29/120	1/15	13/120	41/120	
	1/60	13/120	73/120	1/20	1/12	2/15	
	1/120	7/20	43/120	7/120	11/120	2/15	
	1/120	3/10	49/120	7/120	1/12	17/120	
	1/120	4/15	53/120	1/20	11/120	17/120	
	1/120	13/60	61/120	1/20	1/12	2/15	
210	1/15	41/210	8/35	1/14	31/210	61/210	$(R_7 : (R_5 : 2R_3))$
	13/210	1/10	83/210	1/14	4/35	9/35	
	1/35	2/15	97/210	1/14	17/210	47/210	
	1/210	3/14	121/210	11/210	1/15	3/35	

It can be checked (using Mathematica) that removing any diagonal from a sporadic configuration of three intersecting diagonals yields a configuration whose denominator is the same or half as much, except that it is possible that removing a diagonal from a three-diagonal configuration of denominator 210 or 60 yields one of denominator 70 or 20, respectively, which proves the desired result for these cases. The additive group generated by 1/6 and the normalized arc lengths of a configuration obtained by removing a diagonal from a configuration corresponding to one of the families of Table 4.1, contains $2t$ where t is the parameter, (as can be verified using Mathematica again), which means that adding that third diagonal can at most double the denominator (and throw in a factor of 3, if it isn't already there). Similarly, it is easily checked (even by hand), that the subgroup generated by the normalized arc lengths of a configuration obtained by removing one of the three diagonals of a configuration corresponding to a trivial solution to (2.2), but with intersection point not the center, contains twice the arc lengths of the original configuration. □

COROLLARY 5.2. *If a configuration of three or more diagonals meeting includes three forming a sporadic configuration, then its denominator is* 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, *or* 420.

Proof. Combine the lemma with the list of denominators of sporadic configurations listed in Table 4.2. □

For $k \geq 4$, a list of $2k$ positive rational numbers summing to 1 arises this way if and only if the lists of length $2k-2$, which would arise by removing the first or second diagonal, actually correspond to $k-1$ intersecting diagonals. Suppose $k = 4$. If we specify the sporadic configuration or parameterized family of configurations that arise when we remove the first or second diagonal, we get a set of linear conditions on the eight arc lengths. Corollary 5.2 tells us that we get a configuration with denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420, if one of these two is sporadic. Using Mathematica to perform this computation for the rest of the possibilities in Theorem 4.4 shows that the other four-diagonal configurations, up to rotation and reflection, fall into 12 one-parameter families, which are listed in Table 5.1 by the eight normalized arc lengths and the range for the parameter t, with a finite number of exceptions of denominators among 12, 18, 24, 30, 36, 42, 48, 60, 84, and 120.

We will use a similar argument when $k = 5$. Any five-diagonal configuration containing a sporadic three-diagonal configuration will again have denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420. Any other five-diagonal configuration containing one of the exceptional four-diagonal configurations will have denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 96, 120, 168, and 240, by Lemma 5.1. Finally, another Mathematica computation shows that the one-parameter families of four-diagonal configurations overlap to produce the one-parameter families listed (up to rotation and reflection) in Table 5.2, and a finite number of exceptions of denominators among 18, 24, and 30.

For $k = 6$, any six-diagonal configuration containing a sporadic three-diagonal configuration will again have denominator among 30, 42, 60, 84, 90, 120, 168, 180, 210, 240, and 420. Any six-diagonal configuration containing one of the exceptional four-diagonal configurations will have denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 96, 120, 168, and 240. Any six-diagonal configuration containing one of the exceptional five-diagonal configurations will have denominator among 18, 24, 30, 36, 48, and 60. Another Mathematica computation shows that the one-parameter families of five-diagonal configurations cannot combine to give a six-diagonal configuration.

TABLE 5.1
The one-parameter families of four-diagonal configurations.

								Range
t	t	t	$1/6-2t$	$1/6$	$1/3+t$	$1/6$	$1/6-2t$	$0<t<1/12$
t	$1/6-t$	$1/6-t$	$1/6-t$	t	$1/6$	$1/6+t$	$1/6$	$0<t<1/6$
$1/6-4t$	$2t$	t	$3t$	$1/6-4t$	$1/6$	$1/6+t$	$1/3+t$	$0<t<1/24$
$2t$	$1/2-t$	$2t$	$1/6-2t$	t	$1/6-t$	t	$1/6-2t$	$0<t<1/12$
$1/3-4t$	$1/6+t$	$1/2-3t$	$-1/6+4t$	$1/6-2t$	t	$1/6-t$	$-1/6+4t$	$1/24<t<1/12$
$2t$	t	$3t$	$1/6-2t$	$1/6$	$1/6-t$	$1/3-t$	$1/6-2t$	$0<t<1/12$
t	t	$2t$	$1/3-t$	$1/6$	$1/6-t$	$1/6-t$	$1/6-t$	$0<t<1/6$
$1/3-4t$	$1/6$	t	t	$1/6-2t$	$1/3-2t$	$3t$	$3t$	$0<t<1/12$
$2t$	$1/3-2t$	$1/6-t$	$1/6-t$	$1/6$	$1/6$	t	t	$0<t<1/6$
$1/3-4t$	$2t$	t	t	$1/6-2t$	$1/6$	$1/6+t$	$1/6+t$	$0<t<1/12$
$1/3-4t$	$2t$	$1/6-t$	t	$1/6-2t$	$2t$	$1/3-t$	$3t$	$0<t<1/12$
$2t$	$1/6-t$	t	$1/6-t$	t	$1/6-t$	$2t$	$1/2-3t$	$0<t<1/6$

TABLE 5.2
The one-parameter families of five-diagonal configurations.

										Range
t	$2t$	$1/6-2t$	$1/6$	$1/6-t$	$1/6-t$	$1/6$	$1/6-2t$	$2t$	t	$0<t<1/12$
t	$2t$	$1/6-4t$	$1/6$	$1/6+t$	$1/6+t$	$1/6$	$1/6-4t$	$2t$	t	$0<t<1/24$
t	$1/6-2t$	$-1/6+4t$	$1/3-4t$	$1/6+t$	$1/6+t$	$1/3-4t$	$-1/6+4t$	$1/6-2t$	t	$1/24<t<1/12$
t	$1/6-2t$	$2t$	$1/3-4t$	$3t$	$3t$	$1/3-4t$	$2t$	$1/6-2t$	t	$0<t<1/12$

of five-diagonal configurations cannot combine to give a six-diagonal configuration.

Finally for $k \geq 7$, any k-diagonal configuration must contain an exceptional configuration of three, four, or five diagonals, and hence by Lemma 5.1 has denominator among 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, 240, and 420.

We summarize the results of this section in the following.

PROPOSITION 5.3. *The configurations of $k \geq 4$ diagonals meeting at a point not the center, up to rotation and reflection, fall into the one-parameter families listed in Tables* 5.1 *and* 5.2*, with finitely many exceptions (for fixed k) of denominators among* 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, 240, *and* 420*.*

In fact, many of the numbers listed in the proposition do not actually occur as denominators of exceptional configurations. For example, it will turn out that the only denominator greater than 120 that occurs is 210.

6. The formula for intersection points. Let $a_k(n)$ denote the number of points inside the regular n-gon other than the center where exactly k lines meet. Let $b_k(n)$ denote the number of k-tuples of diagonals which meet at a point inside the n-gon other than the center. Each interior point at which exactly m diagonals meet gives rise to $\binom{m}{k}$ such k-tuples, so we have the relationship

$$b_k(n) = \sum_{m \geq k} \binom{m}{k} a_m(n). \tag{6.1}$$

Since every four distinct vertices of the n-gon determine one pair of diagonals which intersect inside, the number of such pairs is exactly $\binom{n}{4}$, but if n is even, then $\binom{n/2}{2}$ of these are pairs which meet at the center, so

$$b_2(n) = \binom{n}{4} - \binom{n/2}{2}\delta_2(n). \tag{6.2}$$

(Recall that $\delta_m(n)$ is defined to be 1 if n is a multiple of m, and 0 otherwise.)

We will use the results of the previous two sections to deduce the form of $b_k(n)$ and then the form of $a_k(n)$. To avoid having to repeat the following, let us make a definition.

DEFINITION 6.1. *A function on integers $n \geq 3$ will be called* tame *if it is a linear combination (with rational coefficients) of the functions n^3, n^2, n, 1, $n^2\delta_2(n)$, $n\delta_2(n)$, $\delta_2(n)$, $\delta_4(n)$, $n\delta_6(n)$, $\delta_6(n)$, $\delta_{12}(n)$, $\delta_{18}(n)$, $\delta_{24}(n)$, $\delta_{24}(n-6)$, $\delta_{30}(n)$, $\delta_{36}(n)$, $\delta_{42}(n)$, $\delta_{48}(n)$, $\delta_{60}(n)$, $\delta_{72}(n)$, $\delta_{84}(n)$, $\delta_{90}(n)$, $\delta_{96}(n)$, $\delta_{120}(n)$, $\delta_{168}(n)$, $\delta_{180}(n)$, $\delta_{210}(n)$, and $\delta_{420}(n)$.*

PROPOSITION 6.2. *For each $k \geq 2$, the function $b_k(n)/n$ on integers $n \geq 3$ is tame.*

Proof. The case $k = 2$ is handled by (6.2), so assume $k \geq 3$. Each list of $2k$ normalized arc lengths, as in section 5, corresponding to a configuration of k diagonals meeting at a point other than the center, considered up to rotation (but not reflection), contributes n to $b_k(n)$. (There are n places to start measuring the arcs from, and these n configurations are distinct, because the corresponding intersection points differ by rotations of multiples of $2\pi/n$, and by assumption they are not at the center.) So $b_k(n)/n$ counts such lists.

Suppose $k = 3$. When n is even, the family of trivial solutions to the trigonometric equation (2.2) has $U = a/n$, $V = b/n$, $W = c/n$, where a, b, and c are positive integers with sum $n/2$, and X, Y, and Z are some permutation of U, V, W. Each permutation gives rise to a two-parameter family of six-long lists of arc lengths, and the number

of lists within each family is the number of partitions of $n/2$ into three positive parts, which is a quadratic polynomial in n. Similarly each family of solutions in Table 4.1 gives rise to a number of one-parameter families of lists, when n is a multiple of 6, each containing $\lceil n/6 \rceil - 1$ or $\lceil n/12 \rceil - 1$ lists. These functions of n (extended to be 0 when 6 does not divide n) are expressible as a linear combination of $n\delta_6(n)$, $\delta_6(n)$, and $\delta_{12}(n)$. Finally, the sporadic solutions to 2.2 give rise to a finite number of lists, having denominators among 30, 42, 60, 84, 90, 120, and 210, so their contribution to $b_3(n)/n$ is a linear combination of $\delta_{30}(n), \ldots, \delta_{210}(n)$.

But these families of lists overlap, so we must use the principle of inclusion-exclusion to count them properly. To show that the result is a tame function, it suffices to show that the number of lists in any intersection of these families is a tame function. When two of the trivial families overlap but do not coincide, they overlap where two of the a, b, and c above are equal, and the corresponding lists lie in one of the one-parameter families $(t, t, t, t, 1/2 - 2t, 1/2 - 2t)$ or $(t, t, t, 1/2 - 2t, t, 1/2 - 2t)$ (with $0 < t < 1/4$), each of which contains $\lceil n/4 \rceil - 1$ lists (for n even). This function of n is a combination of $n\delta_2(n)$, $\delta_2(n)$, and $\delta_4(n)$; hence it is tame. Any other intersection of the infinite families must contain the intersection of two one-parameter families which are among the two above or arise from Table 4.1, and a Mathematica computation shows that such an intersection consists of at most a single list of denominator among 6, 12, 18, 24, and 30. And, of course, any intersection involving a single sporadic list can contain at most that sporadic list. Thus the number of lists within any intersection is a tame function of n. Finally, we must delete the lists which correspond to configurations of diagonals meeting at the center. These are the lists within the trivial two-parameter family $(t, u, 1/2 - t - u, t, u, 1/2 - t - u)$, so their number is also a tame function of n, by the principle of inclusion-exclusion again. Thus $b_3(n)/n$ is tame.

Next suppose $k = 4$. The number of lists within each family listed in Table 5.1, or the reflection of such a family, is (when n is divisible by 6) the number of multiples of $1/n$ strictly between α and β, where the range for the parameter t is $\alpha < t < \beta$. This number is $\lceil \beta n \rceil - 1 - \lfloor \alpha n \rfloor$. Since the table shows that α and β are always multiples of $1/24$, this function of n is expressible as a combination of $n\delta_6(n)$ and a function on multiples of 6 depending only on n mod 24, and the latter can be written as a combination of $\delta_6(n)$, $\delta_{12}(n)$, $\delta_{24}(n)$, and $\delta_{24}(n-6)$, so it is tame. Mathematica shows that when two of these families are not the same, they intersect in at most a single list of denominator among 6, 12, 18, and 24. So these and the exceptions of Proposition 5.3 can be counted by a tame function. Thus, again by the principle of inclusion-exclusion, $b_4(n)/n$ is tame.

The proof for $k = 5$ is identical to that of $k = 4$, using Table 5.2 instead of Table 5.1, and using another Mathematica computation which shows that the intersections of two one-parameter families of lists consist of at most a single list of denominator 24.

The proof for $k \geq 6$ is even simpler, because then there are only the exceptional lists. By Proposition 5.3, $b_k(n)/n$ is a linear combination of $\delta_m(n)$ where m ranges over the possible denominators of exceptional lists listed in the proposition, so it is tame. □

LEMMA 6.3. *A tame function is determined by its values at* $n = 3$, 4, 5, 6, 7, 8, 9, 10, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 84, 90, 96, 120, 168, 180, 210, *and* 420.

Proof. By linearity, it suffices to show that if a tame function f is zero at those

values, then f is the zero linear combination of the functions in the definition of a tame function. The vanishing at $n = 3$, 5, 7, and 9 forces the coefficients of n^3, n^2, n, and 1 to vanish, by Lagrange interpolation. Then comparing the values at $n = 4$ and $n = 10$ shows that the coefficient of $\delta_4(n)$ is zero. The vanishing at $n = 4$, 8, and 10 forces the coefficients of $n^2\delta_2(n)$, $n\delta_2(n)$, and $\delta_2(n)$ to vanish. Comparing the values at $n = 6$ and $n = 54$ shows that the coefficient of $n\delta_6$ is zero. Comparing the values at $n = 6$ and $n = 66$ shows that the coefficient of $\delta_2 4(n - 6)$ is zero.

At this point, we know that $f(n)$ is a combination of $\delta_m(n)$, for $m = 6$, 12, 18, 24, 30, 36, 42, 48, 60, 72, 84, 90, 96, 120, 168, 180, 210, and 420. For each m in turn, $f(m) = 0$ now implies that the coefficient of $\delta_m(n)$ is zero. □

Proof of Theorem 1.1. Computation (see the appendix) shows that the tame function $b_8(n)/n$ vanishes at all the numbers listed in Lemma 6.3. Hence by that lemma, $b_8(n) = 0$ for all n. Thus by (6.1), $a_k(n)$ and $b_k(n)$ are identically zero for all $k \geq 8$ as well.

By reverse induction on k, we can invert (6.1) to express $a_k(n)$ as a linear combination of $b_m(n)$ with $m \geq k$. Hence $a_k(n)/n$ is tame as well for each $k \geq 2$. Computation shows that the equations

$$\begin{aligned}
a_2(n)/n &= (n^3 - 6n^2 + 11n - 6)/24 + (-5n^2 + 46n - 72)/16 \cdot \delta_2(n) \\
&\quad - 9/4 \cdot \delta_4(n) + (-19n + 110)/2 \cdot \delta_6(n) + 54 \cdot \delta_{12}(n) + 84 \cdot \delta_{18}(n) \\
&\quad + 50 \cdot \delta_{24}(n) - 24 \cdot \delta_{30}(n) - 100 \cdot \delta_{42}(n) - 432 \cdot \delta_{60}(n) \\
&\quad - 204 \cdot \delta_{84}(n) - 144 \cdot \delta_{90}(n) - 204 \cdot \delta_{120}(n) - 144 \cdot \delta_{210}(n), \\
a_3(n)/n &= (5n^2 - 48n + 76)/48 \cdot \delta_2(n) + 3/4 \cdot \delta_4(n) + (7n - 38)/6 \cdot \delta_6(n) \\
&\quad - 8 \cdot \delta_{12}(n) - 20 \cdot \delta_{18}(n) - 16 \cdot \delta_{24}(n) - 19 \cdot \delta_{30}(n) + 8 \cdot \delta_{42}(n) \\
&\quad + 68 \cdot \delta_{60}(n) + 60 \cdot \delta_{84}(n) + 48 \cdot \delta_{90}(n) + 60 \cdot \delta_{120}(n) + 48 \cdot \delta_{210}(n), \\
a_4(n)/n &= (7n - 42)/12 \cdot \delta_6(n) - 5/2 \cdot \delta_{12}(n) - 4 \cdot \delta_{18}(n) + 3 \cdot \delta_{24}(n) \\
&\quad + 6 \cdot \delta_{42}(n) + 34 \cdot \delta_{60}(n) - 6 \cdot \delta_{84}(n) - 6 \cdot \delta_{120}(n), \\
a_5(n)/n &= (n - 6)/4 \cdot \delta_6(n) - 3/2 \cdot \delta_{12}(n) - 2 \cdot \delta_{24}(n) + 4 \cdot \delta_{42}(n) \\
&\quad + 6 \cdot \delta_{84}(n) + 6 \cdot \delta_{120}(n), \\
a_6(n)/n &= 4 \cdot \delta_{30}(n) - 4 \cdot \delta_{60}(n), \\
a_7(n)/n &= \delta_{30}(n) + 4 \cdot \delta_{60}(n)
\end{aligned}$$

hold for all the n listed in Lemma 6.3, so the lemma implies that they hold for all $n \geq 3$. These formulas imply the remarks in the introduction about the maximum number of diagonals meeting at an interior point other than the center. Finally,

$$\begin{aligned}
I(n) &= \delta_2(n) + \sum_{k=2}^{\infty} a_k(n) \\
&= \delta_2(n) + \sum_{k=2}^{7} a_k(n),
\end{aligned}$$

which gives the desired formula. (The $\delta_2(n)$ in the expression for $I(n)$ is to account for the center point when n is even, which is the only point not counted by the a_k.) □

7. The formula for regions. We now use the knowledge obtained in the proof of Theorem 1.1 about the number of interior points through which exactly k diagonals pass to calculate the number of regions formed by the diagonals.

Proof of Theorem 1.2. Consider the graph formed from the configuration of a regular n-gon with its diagonals, in which the vertices are the vertices of the n-gon together with the interior intersection points, and the edges are the sides of the n-gon together with the segments that the diagonals cut themselves into. As usual, let V denote the number of vertices of the graph, E the number of edges, and F the number of regions formed, including the region outside the n-gon. We will employ Euler's formula $V - E + F = 2$.

Clearly $V = n + I(n)$. We will count edges by counting their ends, which are $2E$ in number. Each vertex has $n-1$ edge ends, the center (if n is even) has n edge ends, and any other interior point through which exactly k diagonals pass has $2k$ edge ends, so

$$2E = n(n-1) + n\delta_2(n) + \sum_{k=2}^{\infty} 2ka_k(n).$$

So the desired number of regions, not counting the region outside the n-gon, is

$$\begin{aligned} F - 1 &= E - V + 1 \\ &= \left[n(n-1)/2 + n\delta_2(n)/2 + \sum_{k=2}^{\infty} ka_k(n) \right] - [n + I(n)] + 1. \end{aligned}$$

Substitution of the formulas derived in the proof of Theorem 1.1 for $a_k(n)$ and $I(n)$ yields the desired result. □

Appendix. Computations and tables. In Table A.1 we list $I(n), R(n)$, $a_2(n), \ldots, a_7(n)$ for $n = 4, 5, \ldots, 30$. To determine the polynomials listed in Theorem 1.1, more data were needed, especially for $n \equiv 0 \bmod 6$. The largest n for which this was required was 420. For speed and memory conservation, we took advantage of the regular n-gon's rotational symmetry and focused our attention on only $2\pi/n$ radians of the n-gon. The data from this computation are found in Tables A.2 and A.3. Although we only needed to know the values at those n listed in Lemma 6.3, we give a list for $n = 6, 12, \ldots, 420$ so that the nice patterns can be seen.

The numbers in these tables were found by numerically computing (using a C program and 64-bit precision) all possible $\binom{n}{4}$ intersections and sorting them by x-coordinate. We then focused on runs of points with close x-coordinates, looking for points with close y-coordinates.

Several checks were made to eliminate any fears (arising from round-off errors) of distinct points being mistaken as close. First, the C program sent data to Maple which checked that the coordinates of close points agreed to at least 40 decimal places. Second, we verified for each n that close points came in counts of the form $\binom{k}{2}$ (k diagonals meeting at a point give rise to $\binom{k}{2}$ close points. Hence, any run whose length is not of this form indicates a computational error).

A second program was then written and run on a second machine to make the computations completely rigorous. It also found the intersection points numerically, sorted them and looked for close points, but, to be absolutely sure that a pair of close points p_1 and p_2 were actually the same, it checked that for the two pairs of diagonals (l_1, l_2) and (l_3, l_4) determining p_1 and p_2, respectively, the triples l_1, l_2, l_3 and l_1, l_2, l_4

TABLE A.1

A listing of $I(n)$, $R(n)$, and $a_2(n), \ldots, a_7(n)$, $n = 3, 4, \ldots, 30$. Note that, when n is even, $I(n)$ also counts the point in the center.

n	$a_2(n)$	$a_3(n)$	$a_4(n)$	$a_5(n)$	$a_6(n)$	$a_7(n)$	$I(n)$	$R(n)$
3							0	1
4							1	4
5	5						5	11
6	12						13	24
7	35						35	50
8	40	8					49	80
9	126						126	154
10	140	20					161	220
11	330						330	375
12	228	60	12				301	444
13	715						715	781
14	644	112					757	952
15	1365						1365	1456
16	1168	208					1377	1696
17	2380						2380	2500
18	1512	216	54	54			1837	2466
19	3876						3876	4029
20	3360	480					3841	4500
21	5985						5985	6175
22	5280	660					5941	6820
23	8855						8855	9086
24	6144	864	264	24			7297	9024
25	12650						12650	12926
26	11284	1196					12481	13988
27	17550						17550	17875
28	15680	1568					17249	19180
29	23751						23751	24129
30	13800	2250	420	180	120	30	16801	21480

TABLE A.2

The number of intersection points for one piece of the pie (i.e., $2\pi/n$ radians), $n = 6, 12, \ldots, 210$.

n	$\frac{a_2(n)}{n}$	$\frac{a_3(n)}{n}$	$\frac{a_4(n)}{n}$	$\frac{a_5(n)}{n}$	$\frac{a_6(n)}{n}$	$\frac{a_7(n)}{n}$	$\frac{I(n)-1}{n}$
6	2						2
12	19	5	1				25
18	84	12	3	3			102
24	256	36	11	1			304
30	460	75	14	6	4	1	560
36	1179	109	11	6			1305
42	1786	194	27	13			2020
48	3168	220	25	7			3420
54	4722	288	24	12			5046
60	6251	422	63	12		5	6753
66	9172	460	35	15			9682
72	12428	504	35	13			12980
78	15920	642	42	18			16622
84	20007	805	43	28			20883
90	25230	863	45	21	4	1	26164
96	31240	948	53	19			32260
102	37786	1096	56	24			38962
108	45447	1201	53	24			46725
114	53768	1368	63	27			55226
120	62652	1601	95	31		5	64384
126	73676	1658	72	34			75440
132	85319	1825	71	30			87245
138	97990	2002	77	33			100102
144	112100	2136	77	31			114344
150	127070	2345	84	36	4	1	129540
156	143635	2549	85	36			146305
162	161520	2736	87	39			164382
168	180504	3008	95	47			183654
174	201448	3178	98	42			204766
180	223251	3470	129	42		5	226897
186	247562	3630	105	45			251342
192	273144	3844	109	43			277140
198	300294	4092	108	48			304542
204	329171	4357	113	48			333689
210	359556	4661	125	55	4	1	364402

TABLE A.3
The number of intersection points for one piece of the pie (i.e., $2\pi/n$ radians), $n = 216, \ldots, 420$.

n	$\frac{a_2(n)}{n}$	$\frac{a_3(n)}{n}$	$\frac{a_4(n)}{n}$	$\frac{a_5(n)}{n}$	$\frac{a_6(n)}{n}$	$\frac{a_7(n)}{n}$	$\frac{I(n)-1}{n}$
216	392564	4848	119	49			397580
222	426836	5166	126	54			432182
228	463303	5441	127	54			468925
234	501762	5718	129	57			507666
240	541612	6121	165	61		5	547964
246	584782	6340	140	60			591322
252	629399	6693	137	70			636299
258	676580	6972	147	63			683762
264	725976	7276	151	61			733464
270	777420	7643	150	66	4	1	785284
276	831575	7969	155	66			839765
282	887986	8326	161	69			896542
288	947132	8640	161	67			956000
294	1008358	9056	174	76			1017664
300	1072171	9462	203	72		5	1081913
306	1139436	9780	171	75			1149462
312	1208944	10164	179	73			1219360
318	1281100	10582	182	78			1291942
324	1356315	10957	179	78			1367529
330	1434110	11375	189	81	4	1	1445760
336	1514816	11856	193	89			1526954
342	1598970	12216	192	84			1611462
348	1685843	12661	197	84			1698785
354	1775788	13108	203	87			1789186
360	1868312	13669	231	91		5	1882308
366	1965272	14010	210	90			1979582
372	2064919	14465	211	90			2079685
378	2167754	14930	219	97			2183000
384	2274136	15396	221	91			2289844
390	2383690	15885	224	96	4	1	2399900
396	2496999	16369	221	96			2513685
402	2613536	16896	231	99			2630762
408	2733888	17380	235	97			2751600
414	2857752	17898	234	102			2875986
420	2984383	18598	273	112		5	3003371

each divided the circle into arcs of lengths consistent with Theorem 4.4. Since this test involves only comparing rational numbers, it could be performed exactly.

A word should also be said concerning limiting the search to $2\pi/n$ radians of the n-gon. Both programs looked at slightly smaller slices of the n-gon to avoid problems caused by points near the boundary. We further subdivided this region into twenty smaller pieces to make the task of sorting the intersection points manageable. More precisely, we limited our search to points whose angle with the origin fell between $[c_1 + 2\pi(m-1)/(20n) + \varepsilon, c_1 + 2\pi m/(20n) - \varepsilon)$, $m = 1, 2, \ldots 20$, and also made sure not to include the origin in the count. Here ε was chosen to be .00000000001 and c_1 was chosen to be .00000123 ($c_1 = 0$ would have led to problems since there are many intersection points with angle 0 or $2\pi/n$). To make sure that no intersection points were omitted, the number of points found (counting multiplicity) was compared with $\left(\binom{n}{4} - \binom{n/2}{2}\delta_2\right)/n$.

Acknowledgments. We thank Joel Spencer and Noga Alon for helpful conversations. Also we thank Jerry Alexanderson, Jeff Lagarias, Hendrik Lenstra, and Gerry Myerson for pointing out to us many of the references below.

REFERENCES

[1] G. Bol, *Beantwoording van prijsvraag no.* 17, Nieuw Archief voor Wiskunde, 18 (1936), pp. 14–66.

[2] J. H. Conway and A. J. Jones, *Trigonometric diophantine equations (on vanishing sums of roots of unity)*, Acta Arith., 30 (1976), pp. 229–240.

[3] H. T. Croft and M. Fowler, *On a problem of Steinhaus about polygons*, Proc. Camb. Phil. Soc., 57 (1961), pp. 686–688.

[4] H. Harborth, *Diagonalen im regularen n-Eck*, Elem. Math., 24 (1969), pp. 104–109.

[5] H. Harborth, *Number of intersections of diagonals in regular n-gons*, in Combinatorial Structures and their Applications, Proc. Calgary Internat. Conf. on Combinatorial Structures, Calgary, Alberta, 1969, R. Guy, H. Hanani, N. Sauer, and J Schöheim, eds., Gordon and Breach, New York, 1970, pp. 151–153.

[6] H. Heineken, *Regelmässige Vielecke und ihre Diagonalen*, Enseign. Math. (2), sér. 8 (1962), pp. 275–278.

[7] H. Heineken, *Regelmässige Vielecke und ihre Diagonalen II*, Rend. Sem. Mat. Univ. Padova, 41 (1968), pp. 332–344.

[8] H. Mann, *On linear relations between roots of unity*, Mathematika, 12 (1965), pp. 107–117.

[9] G. Myerson, *Rational products of sines of rational angles*, Aequationes Math., 45 (1993), pp. 70–82.

[10] J. F. Rigby, *Adventitious quadrangles: A geometrical approach*, Math. Gaz., 62 (1978), pp. 183–191.

[11] J. F. Rigby, *Multiple intersections of diagonals of regular polygons, and related topics*, Geom. Dedicata, 9 (1980), pp. 207–238.

[12] I. J. Schoenberg, *A note on the cyclotomic polynomial*, Mathematika, 11 (1964), pp. 131–136.

[13] H. Steinhaus, *Mathematical Snapshots*, Oxford University Press, Oxford, 1983, pp. 259–260.

[14] H. Steinhaus, *Problem* 225, Colloq. Math., 5 (1958), p. 235.

[15] C. E. Tripp, *Adventitious angles*, Math. Gaz., 59 (1975), pp. 98–106.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 157–167, February 1998

011

GEOMETRY AND DIAMETER BOUNDS OF DIRECTED CAYLEY GRAPHS OF ABELIAN GROUPS*

CHARLES M. FIDUCCIA†, RODNEY W. FORCADE‡, AND JENNIFER S. ZITO†

Abstract. Many popular interconnection network topologies, such as hypercubes and toroidal meshes, are based on Cayley graphs of Abelian groups. The symmetry and algebraic structure of these graphs result in many nice physical properties of the network concerning layout, routing algorithms, and load balancing. There has been interest in low-diameter Abelian–Cayley graphs because of their smaller communication delay and reduced congestion. For any fixed number of nodes n, and any fixed out-degree k, we are interested in how small the diameter of directed Cayley graphs of Abelian groups can be and what these low-diameter graphs look like. We give an upper bound of $n \leq \frac{3(d+3)^3}{25}$ for the size of directed Abelian–Cayley graphs with $k = 3$ and diameter d, correcting a previously published result by Hsu and Jia [*SIAM J. Discrete Math.*, 7 (1994), pp. 57–71].

Our method is based on translational tiling techniques and is a generalization of Wong and Coppersmith's method for $k = 2$ [*J. Assoc. Comput. Mach.*, 21 (1974), pp. 392–402]. Moreover, our method works for all Abelian groups, not just the cyclic case. For $k = 3$ we give computational results for the largest Abelian–Cayley graph as a function of diameter. When $n = 84m^3$, for integer m, there is a network with $n = \frac{(d+3)^3}{11.95}$ whose diameter is approximately three-fourths of that of a three-dimensional toroidal cube.

Key words. Cayley graphs, Abelian groups, diameter, network topologies, translational tilings, degree-diameter problems

AMS subject classifications. 05C25, 05C12, 20F05, 68R10, 90B12

PII. S0895480195286456

1. Introduction and problem history.

The directed Cayley graph associated with an Abelian group G and an edge generating set $E \subset G$ has the elements of G as its vertices and directed edges from each g to all vertices $g' = g + h$, where $h \in E$. The out-degree k of the graph, as well as its in-degree, is thus equal to the cardinality of E.

The process of moving through a Cayley graph can be geometrically represented by associating each generator of the group with an orthogonal direction (i.e., a unit vector) in k-dimensional space, starting with the group identity at the origin. Thus each edge-generator of the Cayley graph corresponds to a direction in the lattice. Given the Cayley graph of an Abelian group G, with n elements and edge-generating list $E = [g_1, g_2, \ldots, g_k]$, we may define a mapping ϕ from the k-dimensional lattice of integers to G. The map $\phi : Z^k \to G$ takes the lattice point with coordinates $(x_1, x_2, \ldots, x_k)$ to the group element $g = x_1 g_1 + x_2 g_2 + \cdots + x_k g_k$ and is a group homomorphism.

Now we do a breadth-first search in the positive orthant of the lattice Z^k, starting at the origin, until we find all the elements of the group G. We do the search in short-lex order with respect to the k dimensions. *Short-lex ordering* for lattice points is defined as follows: the lattice point $\vec{x} = (x_1, x_2, \ldots, x_k)$ precedes the lattice point $\vec{y} = (y_1, y_2, \ldots, y_k)$ if either $\vec{x}$ has smaller *Manhattan distance* from the origin ($\sum_{i=1}^{k} x_i <$

* Received by the editors May 24, 1995; accepted for publication (in revised form) December 5, 1996.

http://www.siam.org/journals/sidma/11-1/28645.html

† Center for Computing Science, 17100 Science Drive, Bowie, MD 20715 (fiduccia@super.org, zito@super.org).

‡ Department of Mathematics, Brigham Young University, Provo, UT 84602 (forcader@math.byu.edu).

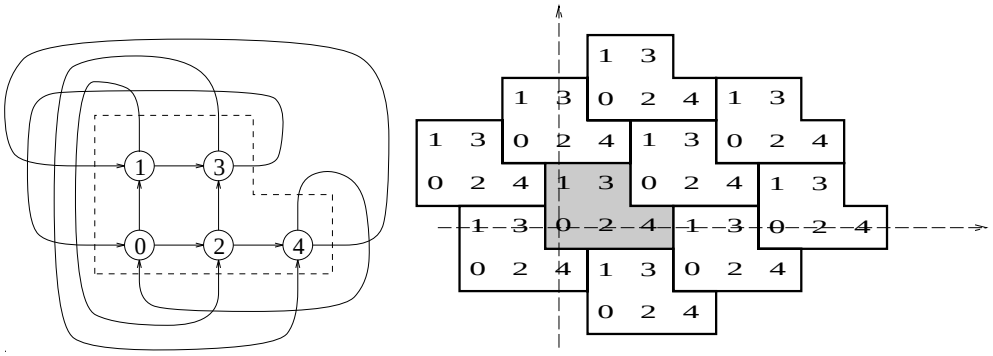


FIG. 1. *The relationship between the Cayley graph of the group $G = Z_5$, with edge-generating list $E = [2, 1]$, the Cayley tile (shaded), and the corresponding tiling of the integer lattice.*

$\sum_{i=1}^{k} y_i$) or $\vec{x}$ and $\vec{y}$ have equal Manhattan distance from the origin and $\vec{x}$ precedes $\vec{y}$ lexicographically. If $g \in G$, let ShortLex(g) be the first lattice point $\vec{x}$ found during the short-lex search such that $\phi(\vec{x}) = g$. Now let $L = \{\text{ShortLex}(g) : g \in G\}$. The set L consists of the n lattice locations for the elements of G as they are first traversed in the short-lex search. Because ϕ is a one-to-one map of L onto G, the elements of L form a complete set of coset representatives (a transversal) for the subgroup $H = ker(\phi)$ (the kernel). In other words, each element z of Z^k is uniquely represented as a sum $z = x + h$, where $x \in L$ and $h \in H$, which is another way of saying that L *tiles* Z^k via the translation group H, that is, $Z^k = L + H$.

Note that the diameter of G corresponds to the greatest Manhattan distance of any point in L from the origin. We will accordingly call it the *diameter* of L.

Now we form a solid tile from our set of lattice points L by taking the union (in R^k)

$$T = L + [0, 1)^k = \bigcup_{x \in L} (x + [0, 1)^k)$$

of the unit k-cubes located at the lattice points of L. This forms a k-dimensional connected shape which we will call the *Cayley tile* at the origin. An example is given in Figure 1.

Note that a tiling of Z^k by L, with the translation group H, corresponds to a tiling of R^k by T, using the same translation group. Note also that, if d is the diameter of L, as defined above, then $d + k$ is the greatest Manhattan distance of any point of the closure of T from the origin. We will call $D = d + k$ the *diameter* of T. (To avoid confusion, we will consistently use D for the latter (solid) diameter and d for the former (lattice) diameter.)

The following example is the Cayley tile created by taking the cyclic group Z_{84}, with edge-generating list $E = [2, 9, 35]$. This tile was found via computer search by one of our summer students, Wei-Hwa Huang, in 1993 and independently by Randall Dougherty and Vance Faber [4]. Readers interested in Cayley graphs as network topologies may also like to see some of the seminal papers in the area [1, 2, 3, 6, 7, 9].

2. Necessary condition on the three-dimensional tiles. The Cayley tile T has several useful properties. If $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$ are elements of R^k, we will say $x \preceq y$ when $x_i \leq y_i$, for all i. Let (0) denote the origin in R^k. Let $\hat{e}_i$ denote the ith unit vector in R^k. Then a *notch* in T is a point x in the

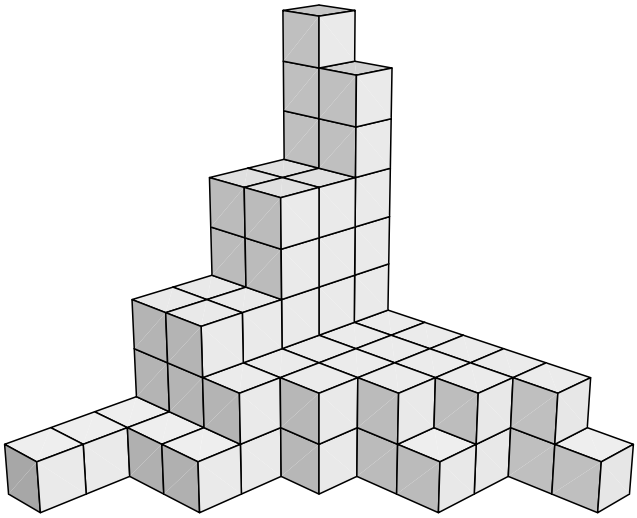


FIG. 2. *The Cayley tile generated by short-lex search of* $G = Z_{84}$ *with edge-generating list* $E = [2, 9, 35]$. *Here the* x_1, x_2, *and* x_3 *axes are to the left, right, and up, respectively.*

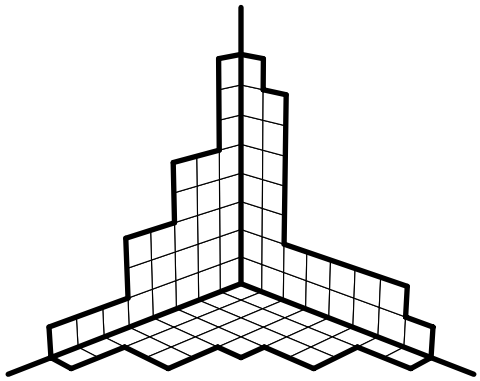


FIG. 3. *The silhouette of the Cayley tile in Figure* 2.

boundary of T so that, for $\epsilon > 0$, $x + \epsilon\,\hat{e}_i$ is also in the boundary of T for each i, but $x + \sum_{i=1}^{k} \epsilon\,\hat{e}_i$ is not in the boundary of T, for any $\epsilon > 0$ (for example, the point $(2, 5, 1)$ in Figure 2). In other words, a notch is a place where it looks like a translation of the first orthant has been cut out of the tile. By the *silhouette* of a Cayley tile T, we mean the set of points y, with at most one nonzero coordinate, such that $y \preceq x$ for some point $x \in T$ (in other words, the projection of T into the coordinate hyperplanes). See Figure 3 for an example.

THEOREM 2.1. *Every Cayley tile* T *has the following properties:*

1. *If* $x \in T$ *and* $(0) \preceq y \preceq x$, *then* $y \in T$.
2. T *has at most one notch.*
3. *A Cayley tile* T *is uniquely determined by its silhouette and, if it has a notch, the coordinates of its notch.*

Proof.

1. First, observe that $L = ShortLex(G)$, as defined above, has this property as a subset of Z^k. For if $(0) \preceq y \preceq x$, with $x \in L$ then, either $y = x$, or y is closer (in Manhattan distance) to the origin. In the former case, there is nothing

to prove. In the latter case, let $\phi(x) = g$ and $\phi(y) = h$. Then $y \notin L$ means there exists y', earlier in short-lex order than y, with $\phi(y') = h$. But then $x' = y' + (x - y)$ is earlier in short-lex order than x and $\phi(x') = h + (g - h) = g$, contradicting $x = ShortLex(g)$. Second, observe that the k-cube $[0, 1)^k$ has this property, as a subset of R^k (trivially). Putting these two observations together implies that $T = L + [0, 1)^k$ also has the property.

2. Suppose T has at least two notches. Call them x and y. Clearly they must be elements of Z^k. As already observed, the neighborhood of a notch looks like a neighborhood of the origin in the complement of the first orthant. Thus, in the translational tiling by T, the only way to "fill" that point is with a translation of the origin itself. Thus both notches are images of the identity of G and so is their difference $x - y$. Since they are notches, each point is immediately above a point of Z^k which is in the tile (both T and L). Simply subtract one from the first coordinate of x and y to get x' and y', respectively. Then $x' - y' = x - y$, so x' and y' have the same ϕ-image, contradicting the definition of L.
3. If $x \in T$, then $\pi_i(x)$ is in the silhouette, for each projection π_i. Furthermore, if there is a notch, y, then $y \not\preceq x$. Conversely, we show that if a point x has these two properties, then it is an element of T. Let x be an element of R^k with every projection $\pi_i(x)$ in the silhouette of T, and $x \notin T$. Then, decreasing the coordinates of x, successively, there is a point $y \preceq x$ with $y \notin T$; but every point z with $z \preceq y$ is in T. Clearly, y is a notch. Thus, if x has all its projections in the silhouette and is not preceded by a notch, then $x \in T$. Thus we have shown that the silhouette and (if it exists) the notch of T entirely determine which points are in T. □

These are necessary, but not sufficient, conditions for a shape to be a Cayley tile. We note that Wong and Coppersmith [9] proved the one-notch result (2) for the two-dimensional case with cyclic groups and one generator equal to the identity.

3. Improved diameter bounds. Given a solidified Cayley tile T, with solid diameter D and volume V, we will show that there exists a shape S with the same diameter and which, although it is not a superset of T, necessarily has greater volume than T. The volume of S will be our bound for that of T. To clarify our argument, however, we do it first with a simplifying assumption—that our tile T has no notch. This will give a bound on volume which holds only for tiles with no notch and which is too small to be proved in the general case.

No notch argument. If T has no notch, it is entirely determined by its silhouette. This means that every point outside of T in the first octant is connected to (at least) one of the coordinate planes by a perpendicular to that coordinate plane. This perpendicular does *not* intersect T. We may classify a point $p \notin T$, in the first octant, as being of *type* x, y, or z, according to whether there is a line through p parallel to the x-axis, y-axis, or z-axis, respectively, which does not intersect T. Note that p may be, simultaneously, of more than one type.

LEMMA 3.1. *If $p \preceq p_1$, then p_1 has (at least) the same type(s) as p.*

Proof. Suppose (for example) that p has type x. Then the segment pq, joining p perpendicularly to the nearest point q in the yz plane, does not intersect T (thus $q \notin T$). But, for every point u on the segment p_1q_1 joining p_1 perpendicularly to the point q_1 in the yz plane, $q \preceq u$, so $u \notin T$. Thus p_1 also has type x. Analogous arguments work for the other two types. □

Let P be the plane defined by $x + y + z = D$. Let O denote the origin. Let $A_1 =$

$(D,0,0)$, $A_2 = (0,D,0)$, and $A_3 = (0,0,D)$. Then T is a subset of the tetrahedron $OA_1A_2A_3$, enclosed by the three coordinate planes and P. Let Q denote the triangle $A_1A_2A_3$ (including its interior area).

Let Γ_x denote the subset of Q comprising all points of type x. Let Γ_y denote the subset of Q comprising all points of Q which are *not* in Γ_x and which are of type y. Let Γ_z be the set of all points of Q which are *not* in $\Gamma_x \cup \Gamma_y$ and which are of type z. By our assumption (no notch) $Q = \Gamma_x \cup \Gamma_y \cup \Gamma_z$. We have also arranged that this be a *disjoint* union.

From each point p in Γ_x, one may drop a segment pq_p, parallel to the x-axis, to a point q_p in the yz plane, without intersecting T. The union of all such segments pq_p, $(p \in \Gamma_x)$ forms a solid G_x in the complement of T. Similarly, from each point in Γ_y we drop a perpendicular segment to the xz plane and let G_y be the union of those segments, and from each point in Γ_z drop a perpendicular segment to the xy plane, thus forming G_z.

LEMMA 3.2. *The sets G_x, G_y, and G_z are disjoint.*

Proof. Suppose $r \in G_x \cap G_y$. Then r is on a line segment p_1q_1 from a point $p_1 \in \Gamma_x$ to the yz plane and r is also on a segment p_2q_2 from a point $p_2 \in \Gamma_y$ to the xz plane. Clearly, r has *both* types and $r \preceq p_1$ and $r \preceq p_2$. By Lemma 1, p_2 is therefore of type x and should therefore have already been included in Γ_x. It cannot be in Γ_y, by our definition. Analogous arguments preclude any other intersection among the three sets. □

Since T is a subset of the tetrahedron $OA_1A_2A_3$, and disjoint from the (disjoint) union $G_x \cup G_y \cup G_z$, it now follows that

$$vol(T) \leq D^3/6 - vol(G_x) - vol(G_y) - vol(G_z).$$

Notice that

$$vol(G_x) + vol(G_y) + vol(G_z) = \int\int_{p \in Q} \lambda\, \delta(p)\, da,$$

where da denotes the differential of area, $\lambda = \frac{1}{\sqrt{3}}$ is a constant introduced because we are integrating over the slanted plane P instead of over the coordinate planes, and $\delta(p)$ is the distance from p to an appropriate coordinate plane. If $p \in \Gamma_x$ then $\delta(p)$ is the distance from p to the yz plane; if $p \in \Gamma_y$, then $\delta(p)$ denotes the distance from p to the xz plane, etc.

A lower bound for this double integral over the triangle Q will thus provide an upper bound for the volume of T. The integral will be smallest when the integrand $\delta(p)$ is as small as possible at every point, and that will be true if $\delta(p) = \delta_1(p)$, where $\delta_1(p)$ is the distance from p to the *nearest* coordinate plane. Thus,

$$vol(T) \leq \frac{D^3}{6} - \int\int_{p \in Q} \lambda\, \delta_1(p)\, da.$$

The right side of this inequality can be explicitly integrated, with some difficulty, but it is more easily interpreted as the volume of a star-shaped (Figure 4) object formed by adjoining to the cube $C = [0, \frac{D}{3})^3$ three pyramids slanting from the faces of C to the points $(D,0,0)$, $(0,D,0)$, and $(0,0,D)$, respectively. Its volume is one-ninth times D^3. Thus, in the case that T has no notch, we have shown that

$$vol(T) \leq \frac{D^3}{9}.$$

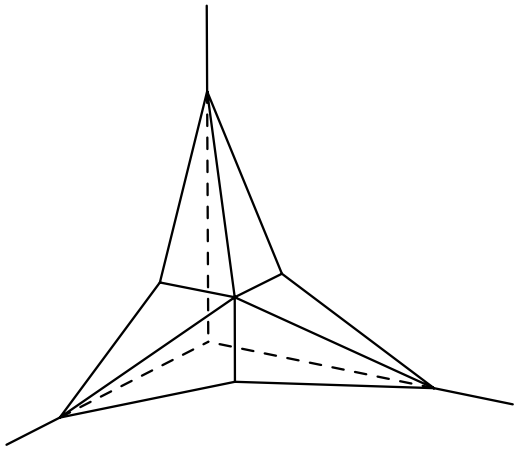


FIG. 4. *Bounding shape for Cayley tiles with no notches.*

With notch argument. How does the argument change if T has a notch? Then, not all of the points on Q are of type x, y, or z; but those which aren't must be able to "see" the notch. If n is the notch, let Δ be the set of points $p \in Q$ such that $n \preceq p$. Following the previous argument, let Γ_x denote the subset of $Q \setminus \Delta$ comprising all points of type x. Let Γ_y denote the subset of $Q \setminus \Delta$ comprising all points which are *not* in Γ_x and which are of type y. Let Γ_z be the set of all points of $Q \setminus \Delta$ which are *not* in $\Gamma_x \cup \Gamma_y$ and which are of type z (see Figure 5). Then Q is the disjoint union of Γ_x, Γ_y, Γ_z, and Δ.

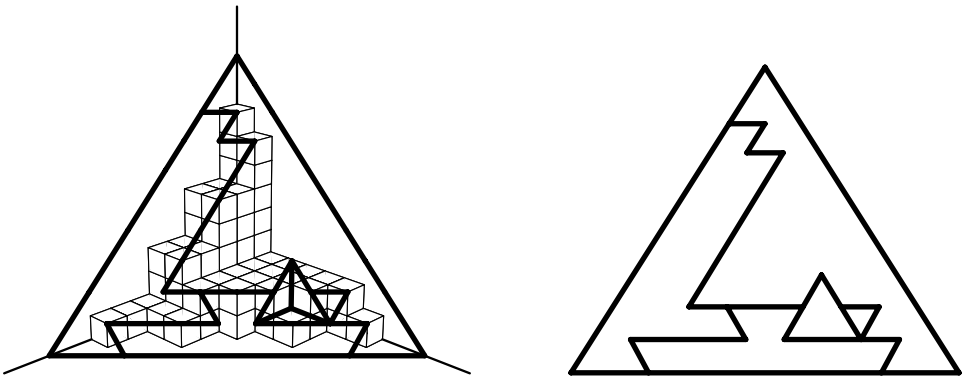


FIG. 5. *Construction of the* Γ *regions.*

Again, let G_x be the union of all segments from Γ_x perpendicular to the yz plane; let G_y be the union of all segments from Γ_y perpendicular to the xz plane; and let G_z be the union of all segments from Γ_z perpendicular to the xy plane. The argument of Lemma 3.2 still works, proving that G_x, G_y, and G_z are disjoint. Let H denote the union of all segments from Δ to the notch point n. Thus,

$$vol(T) \leq D^3/6 - vol(G_x) - vol(G_y) - vol(G_z) - vol(H).$$

Now

$$vol(G_x) + vol(G_y) + vol(G_z) = \int \int_{p \in Q \setminus \Delta} \lambda \, \delta(p) \, da,$$

where da denotes the differential of area, $\lambda = \frac{1}{\sqrt{3}}$ is a constant introduced because we are integrating over the slanted plane P instead of over the coordinate planes, and $\delta(p)$ is the distance from p to an appropriate coordinate plane (for elements in Γ_x, Γ_y, or Γ_z).

In fact, by introducing another constant, λ',

$$vol(G_x) + vol(G_y) + vol(G_z) = \int\int_{p \in Q \setminus \Delta} \lambda' \, \delta'(p) \, da,$$

where $\delta'(p)$ is the distance from p to the intersection of Q with one of the three coordinate planes (depending on which region, Γ_x, Γ_y, or Γ_z p is in).

Clearly, this integral will be made smaller if the Γ regions are adjusted so that $\delta'(p)$ is always the distance from p to the *nearest* edge of Q when $p \notin \Delta$ (see the first diagram in Figure 6).

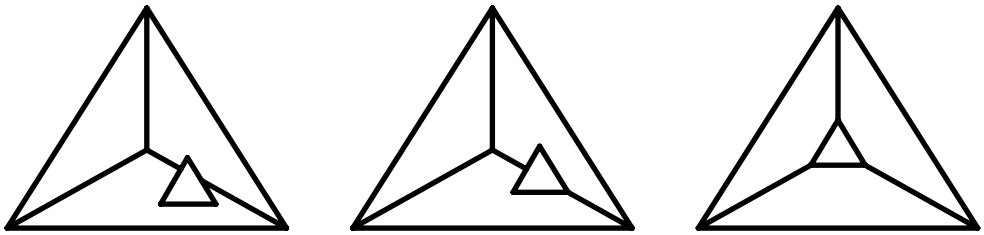


FIG. 6. *Placement of the delta.*

Thus, we may assume that the (new) Γ regions are bounded by the three lines from the vertices of Q to its center. For convenience, let us refer to those lines as the *propeller* lines. The only question remaining is where and how big Δ should be, in order to minimize the integral

$$I \;=\; \int\int_{p \in Q \setminus \Delta} \lambda' \, \delta_1(p) \, da,$$

where $\delta_1(p)$ is the distance from p to the nearest side of Q.

Note that Q and Δ are equilateral triangles with parallel sides. Note also that if none of the vertices of Δ is on a propeller line, then at least one edge, E, of Δ lies entirely within the region bounded by the propellers and by the edge of Q parallel to it (for if each edge crosses a propeller, it meets another edge which is closer to the outside of the triangle, which crosses another propeller and meets...etc.).

LEMMA 3.3. *If we slide Δ, keeping its shape, size, and orientation constant, in a direction perpendicular to and away from that edge of Q which is parallel to E, the double integral I will be decreased.* (*See the first two diagrams of Figure* 6.)

Proof. The differential for I is given by the change in that part which is taken over $Q \setminus \Delta$. Thus,

$$dI \;=\; \left(\int_E \delta_1(p)|dp| - \frac{1}{2}\int_b \delta_1(p)|dp| - \frac{1}{2}\int_c \delta_1(p)|dp| \right) ds,$$

where ds is the differential of distance in the direction implied by the lemma statement, and the $|dp|$ differentials mean that the three single integrals are to be taken over the three sides (E, b, and c) with respect to positive distance along those sides. The

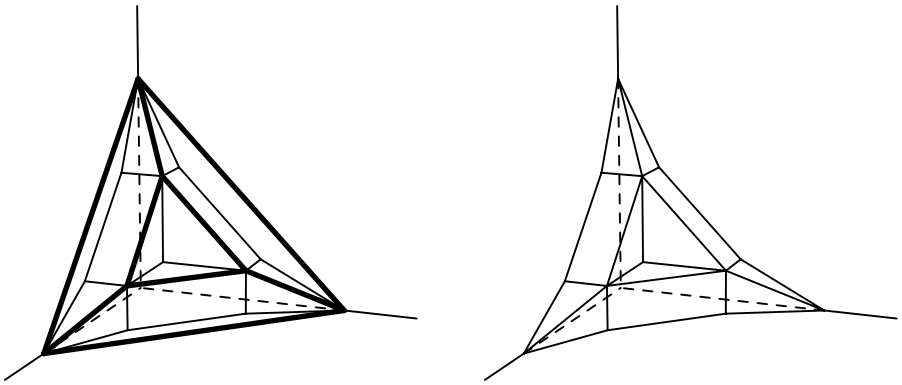


FIG. 7. *Bounding shape for notched tiles.*

factors of $\frac{1}{2}$ come from the cosine of 60 degrees (since the sides b and c are slanted at that angle to E).

Since the distance from every point on b (for instance) is at least as far from the nearest edge of Q as is the entire edge E (at one end of b distances are the same, but points near that end are farther away), the integral over b above is bigger than the one over E. Similarly, the integral over c is bigger than the one over E. Thus, even with the factors of $\frac{1}{2}$, the two negative integrals overpower the positive one, and $dI < 0$. □

LEMMA 3.4. *If one of the vertices of Δ is on a propeller line, and if the opposite edge is moved closer to the position where its ends lie on the propeller lines (keeping the one vertex on the propeller line and keeping the size of Q fixed) the double integral I will decrease. (This is illustrated by the last two diagrams of Figure* 6.*)*

Proof. Letting E be the edge opposite the vertex v on a propeller line, either E is outside of the other two propeller lines, in which case the previous lemma applies, or E lies on the same side of the center of Δ as v (in which case the assertion is rather trivial), or E crosses *both* of the other two propeller lines. Write (as before)

$$dI \;=\; \left(\int_E \delta_1(p)dp - \frac{1}{2}\int_b \delta_1(p)|dp| - \frac{1}{2}\int_c \delta_1(p)|dp|\right)ds,$$

(where b and c are now the two edges emanating from v. Since each of those edges lies entirely in a region bounded by propeller lines, each of the negative integrals in our differential is (strictly) greater than the positive one. Thus a complementary argument (to the proof of the previous lemma) applies. The double integral will be decreased by moving v closer to the center of Q. □

Now, the problem is reduced to a simple calculus problem. Given that the two triangles have a common center, what size Δ maximizes $D^3/6 - I - vol(H)$, which describes a specific shape (Figure 7) with volume

$$u^3 + 3u^2(D-3u) + 3\frac{1}{2}u(D-3u)^2 + 3\frac{1}{3}u^2(2u),$$

where (u,u,u) are the coordinates of the notch n, which defines the size of Δ on Q (the smaller u, the larger Δ is, but centered on Q)? Taking the derivative and setting it equal to zero gives

$$(5u-D)(3u-D) = 0.$$

The root $u = \frac{D}{3}$ makes the notch vanish and gives the value $\frac{D^3}{9}$ for the volume estimate $(\frac{D^3}{6} - I)$. The root $u = \frac{D}{5}$ is a local maximum (i.e., it corresponds to a minimum of our double integral) and gives the value $\frac{3D^3}{25}$ for our volume estimate. Thus, we have the following.

THEOREM 3.5. *If the solid diameter of a three-dimensional Cayley tile T is D, then the volume of T is at most $\frac{3D^3}{25}$.*

COROLLARY 3.6. *If the Cayley graph of a finite Abelian group G with three generators has diameter d, then G has at most $\frac{3(d+3)^3}{25}$ elements.*

Hsu and Jia [6] claimed to show that $n \leq \frac{(d+3)^3}{8.8}$, which would have been a better bound than the one proved in our paper; however, their proof is flawed. Discussion of the proof of their Theorem 3 is made more difficult by the vagueness of their assumptions. Although they do not state this clearly, it appears that they understood that a tile is uniquely determined by its silhouette and the position of its notch, if it has one. Apparently they believed that in order to maximize the volume of a tile-like shape, all the extreme points of its silhouette must lie along a triangle. However, as we can see in our Figure 7, the silhouette can be more elongated along the axes.

Their paper does, however, contain a beautiful constructive proof that the volume of the largest Cayley tile for a *cyclic* group with three generators, with a given diameter d, is at least $\frac{d^3}{16}$ asymptotically.

4. Computational results. The following table contains the largest number of nodes n such that there exists an Abelian group on three generators with diameter d for d up to 17. Table 1 includes the value α such that $n = \alpha(d+3)^3 = \alpha D^3$. Notice that our bound has value $\alpha = 3/25 = 0.12$ and the best value for actual Cayley graphs in the table below is when $n = 84$, which has $\alpha = 84/(10^3) = .084$. The table also compares the solid diameter D to the diameter that a cube of the same volume would have. This is expressed by the number β given by $D = \beta(3\sqrt[3]{n})$. In the table (for each diameter d and corresponding largest number of nodes n), we give the generators for the first Abelian–Cayley graph found by our computer search. For all cyclic cases, with the exception of $d = 7$, there was an Abelian–Cayley graph of minimal diameter that had 1 as a generator.

The case $d = 17$ shows that the best diameter for $n = 672 = 84 \times 2^3$ was obtained by taking the tile for $n = 84$ and replacing every unit cube by a $2 \times 2 \times 2$ cube. One strategy for possibly improving on the best α and β is to take the n which achieves these best values and look at the values of α and β obtained from multiples $n \times m^3$, for $m = 2, 3, \ldots$. Since each of these requires a great deal of computer time, only the first several cases, $n = 2268 = 84 \times 3^3$ and $n = 5376 = 84 \times 4^3$, were attempted. Neither case produced better values of α and β.

5. Infinite families of tiles. We can create infinite families of tiles by scaling up existing tiles. Each cube of the smaller tile is replaced by an m_1 by m_2 by m_3 block of cubes. The group of the scaled-up tile will not be cyclic in the case when $\gcd(m_1, m_2, m_3) \neq 1$. The group of the scaled-up tile can be calculated. In fact, it is possible to find the Cayley group and a set of generators of the tile given a translational tiling via Smith Normal form [5]. If this block is cubical ($m = m_1 = m_2 = m_3$) and the original tile had volume n and diameter $D = d+3$, then the scaled up version has volume $m^3 n$ and diameter mD and hence retains the same values of α and β. The group of the scaled-up tile will not be cyclic if $m > 1$. The scaled-up version of the *tripod* shaped tile in the table for $d = 1$ gives a family which is a special case of what

TABLE 1

d	n_{max}	Group	Generators			α	β
1	4	Z_4	1	2	3	0.06250	0.83995
2	9	Z_9	1	3	4	0.07200	0.80125
3	16	Z_{16}	1	4	5	0.07407	0.79370
4	27	Z_{27}	1	4	17	0.07872	0.77778
5	40	Z_{40}	1	6	15	0.07813	0.77974
6	57	Z_{57}	1	13	33	0.07819	0.77952
7	84	Z_{84}	2	9	35	0.08400	0.76112
8	111	Z_{111}	1	31	69	0.08340	0.76295
9	138	Z_{138}	1	11	78	0.07986	0.77405
10	176	Z_{176}	1	17	56	0.08011	0.77325
11	217	Z_{217}	1	13	119	0.07908	0.77658
12	279	$Z_{93}\times Z_3$	(1,0)	(4,1)	(59,1)	0.08267	0.76519
13	340	Z_{340}	1	90	191	0.08301	0.76414
14	395	Z_{395}	1	35	271	0.08040	0.77232
15	462	Z_{462}	1	29	97	0.07922	0.77614
16	560	Z_{560}	1	215	326	0.08164	0.76837
17	672	$Z_{168}\times Z_2\times Z_2$	(2,0,1)	(9,0,0)	(35,1,0)	0.08400	0.76112

Sherman Stein called semicrosses [8]. One of the nice properties of scaling is that the simplicity of the shape is preserved.

There are also infinite families of Cayley tiles which do not arise from scaling. For example, a family that we call the *double staircases* has $n = k^2$ nodes, diameter $k-1$, and edge generating list $[1, k, k+1]$ (see Figure 8 for $k = 9$, $n = 81$). The entry in the table above for $d = 3$ is a double staircase.

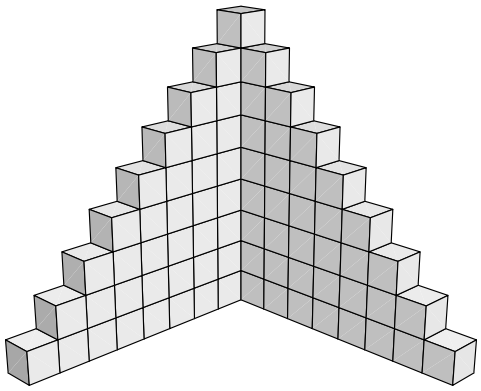


FIG. 8. *Double staircase.*

In terms of evaluating the Abelian–Cayley graphs as network topologies, some of these families of tiles have a useful regularity of shape which carries over into physical layout and routing algorithms. Even though the tile in the table above with $n = 84$ has the lowest diameter as a function of volume, it lacks such regularity. The double staircases and the scaled tripods are two families which manage to combine regularity and low diameter. We have results concerning other families of shapes which will appear in future paper.

6. Open problems. In two dimensions, Wong and Coppersmith [9] showed that a necessary and sufficient condition for a shape to be a tile is that it be L-shaped. In [5], Fiduccia, Zito, and Mann give some conditions on a three-dimensional shape that are necessarily satisfied if the shape is a Cayley tile. However, no set of neces-

sary and sufficient conditions are known for the three-dimensional case. Even less is known about the higher-dimensional cases. What are best diameter tiles in higher dimensions? Can the methods of this paper be generalized to four dimensions and higher? Can the diameter bound in three dimensions be improved?

REFERENCES

[1] S. B. Akers and B. Krishnamurthy, *A group-theoretic model for symmetric interconnection networks*, IEEE Trans. Comput., 38 (1989), pp. 555–566.

[2] F. Annexstein and M. Baumslag, *On the diameter and bisector size of Cayley graphs*, Math. Systems Theory, 26 (1993), pp. 271–291.

[3] G. Cooperman and L. Finkelstein, *New methods for using Cayley graphs in interconnection networks*, Discrete Appl. Math., 37/38 (1992), pp. 95–118.

[4] R. Dougherty and V. Faber, *The degree-diameter problem for several varieties of Cayley graphs, I: The Abelian Case*, Los Alamos National Laboratory E-print Server, http://www.c3.lanl.gov/dm/pub/laces.html (1994).

[5] C. M. Fiduccia, J. S. Zito, and E. Mann, *Network Interconnection Architectures and Translational Tilings*, Technical Report, Center for Computing Science, Bowie, MD, 1994.

[6] D. F. Hsu and X. D. Jia, *Extremal problems in the construction of distributed loop networks*, SIAM J. Discrete Math., 7 (1994), pp. 57–71.

[7] F. D. Hwang and Y. H. Xu, *Double loop networks with minimum delay*, Discrete Math., 66 (1987), pp. 109–118.

[8] S. K. Stein, *Algebraic tiling*, Amer. Math. Monthly, 81 (1974), pp. 445–462.

[9] C. K. Wong and D. Coppersmith, *A combinatorial problem related to multimodule memory organizations*, J. Assoc. Comput. Mach., 21 (1974), pp. 392–402.

SIAM J. DISCRETE MATH.
Vol. 11, No. 1, pp. 168–181, February 1998

012

RANKINGS OF GRAPHS*

HANS L. BODLAENDER†, JITENDER S. DEOGUN‡, KLAUS JANSEN§, TON KLOKS¶, DIETER KRATSCH‖, HAIKO MÜLLER**, AND ZSOLT TUZA††

Abstract. A vertex (edge) coloring $\phi : V \to \{1, 2, \ldots, t\}$ ($\phi' : E \to \{1, 2, \ldots, t\}$) of a graph $G = (V, E)$ is a vertex (edge) *t-ranking* if, for any two vertices (edges) of the same color, every path between them contains a vertex (edge) of larger color. The *vertex ranking number* $\chi_r(G)$ (*edge ranking number* $\chi'_r(G)$) is the smallest value of t such that G has a vertex (edge) t-ranking. In this paper we study the algorithmic complexity of the VERTEX RANKING and EDGE RANKING problems. It is shown that $\chi_r(G)$ can be computed in polynomial time when restricted to graphs with treewidth at most k for any fixed k. We characterize the graphs where the vertex ranking number χ_r and the chromatic number χ coincide on all induced subgraphs, show that $\chi_r(G) = \chi(G)$ implies $\chi(G) = \omega(G)$ (largest clique size), and give a formula for $\chi'_r(K_n)$.

Key words. ranking of graphs, vertex ranking, edge ranking, graph algorithms, treewidth, graph coloring

AMS subject classifications. 68R10, 05C85, 05C15, 05C78

PII. S0895480195282550

1. Introduction.

In this paper we consider vertex rankings and edge rankings of graphs. The vertex ranking problem, also called the *ordered coloring problem* [15], has received much attention lately because of the growing number of applications. There are applications in scheduling problems of assembly steps in manufacturing systems [19], e.g., edge ranking of trees can be used to model the parallel assembly of a product from its components in a quite natural manner [6, 13, 14]. Furthermore, the problem of finding an optimal vertex ranking is equivalent to the problem of finding a minimum-height elimination tree of a graph [6, 8]. This measure is of importance for the parallel Cholesky factorization of matrices [3, 10, 18]. Other applications lie in the field of VLSI-layout [17, 26].

The VERTEX RANKING problem. "Given a graph G and a positive integer t, decide whether $\chi_r(G) \leq t$" is NP-complete even when restricted to cobipartite graphs since

* Received by the editors March 6, 1995; accepted for publication (in revised form) January 6, 1997.

http://www.siam.org/journals/sidma/11-1/28255.html

† Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands (hansb@cs.ruu.nl). The research of this author was partially supported by the ESPRIT Basic Research Actions of the EC under contract 7141 (project ALCOM II).

‡ Department of Computer Science and Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588-0115 (jdeogun@cse.unl.edu). The research of this author was partially supported by Office of Naval Research grant N0014-91-J-1693.

§ Max-Planck Institute for Computer Science, Im Stadtwald, 66123 Saarbrücken, Germany (jansen@ mpi-sb.mpg.de). The work of this author was done while he was at Fachbereich IV Mathematik, Universität Trier, Germany.

¶ Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands (ton@win.tue.nl).

‖ Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität, Universitätshochhaus, 07740 Jena, Germany (dieter.kratsch@mathematik.uni-jena.dpb.de). The research of this author was partially supported by Deutsche Forschungsgemeinschaft under contract Kr 1371/1-1.

** Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität, Universitätshochhaus, 07740 Jena, Germany (hm@minet.uni-jena.de).

†† Computer and Automation Institute, Hungarian Academy of Sciences, Kende u. 13–17, H-1111 Budapest, Hungary. The research of this author was partially supported by Hungarian Scientific Research Fund OTKA grants 2569 and T-016416.

Pothen has shown that the equivalent minimum elimination tree height problem remains NP-complete on cobipartite graphs [20]. A short proof of the NP-completeness of VERTEX RANKING is given in section 3. Much work has been done in finding optimal rankings of trees. For trees there is a linear-time algorithm for finding an optimal vertex ranking [24]. For the closely related edge ranking problem on trees an $O(n^3)$ algorithm was claimed in [9], but in [7] some flaws in this algorithm were pointed out. Recently, Zhou, Kashem, and Nishizeki obtained an $O(n^2 \log \Delta)$ algorithm for edge ranking trees optimally where Δ is the maximum degree of the tree under consideration [28]. Efficient vertex ranking algorithms for permutation, trapezoid, interval, circular-arc, circular permutation graphs, and cocomparability graphs of bounded dimension are presented in [8]. Moreover, the vertex ranking problem is trivial on split graphs and it is solvable in linear time on cographs [25].

In [15], typical graph theoretic questions, as they are known from the coloring theory of graphs, are investigated. This also leads to an $O(\sqrt{n})$ bound for the vertex ranking number of a planar graph and the authors describe a polynomial-time algorithm which finds a vertex ranking of a planar graph using only $O(\sqrt{n})$ colors. For graphs in general there is an approximation algorithm of performance ratio $O(\log^2 n)$ for the vertex ranking number [3, 16]. In [3] it is also shown that one, plus the pathwidth of a graph is a lower bound for the vertex ranking number of the graph (hence, a planar graph has pathwidth $O(\sqrt{n})$, which is also shown in [16] using different methods).

Our goal is to extend the known results in both the algorithmic and graph theoretic directions. The paper is organized as follows. In section 2 the necessary notions and preliminary results are given. We study the algorithmic complexity of determining whether a graph G fulfills $\chi_r(G) \leq t$ and $\chi'_r(G) \leq t$, respectively, in sections 3, 4, and 5. In section 6 we characterize the graphs for which the vertex ranking number and the chromatic number coincide on every induced subgraph. These graphs turn out to be precisely the graphs containing no induced path or cycle on four vertices; hence, we obtain a characterization of the *trivially perfect graphs* [12] in terms of rankings. Moreover, we show that $\chi(G) = \chi_r(G)$ implies that the chromatic number of G is equal to its largest clique size. In section 7 we give a recurrence relation allowing us to compute the edge ranking number of a complete graph.

2. Preliminaries. We consider only finite, undirected and simple graphs $G = (V, E)$. Throughout the paper, n denotes the cardinality of the vertex set V and m denotes that of the edge set E of the graph $G = (V, E)$. For graph theoretic concepts, definitions, and properties of graph classes not given here we refer the reader to [4, 5, 12].

Let $G = (V, E)$ be a graph. A subset $U \subseteq V$ is *independent* if each pair of vertices $u, v \in U$ is nonadjacent. A graph $G = (V, E)$ is *bipartite* if there is a partition of V into two independent sets A and B. The *complement* of the graph $G = (V, E)$ is the graph $\overline{G}$ having vertex set V and edge set $\{\{v, w\} \mid v \neq w,\ \{v, w\} \notin E\}$. For $W \subseteq V$ we denote by $G[W]$ the subgraph of $G = (V, E)$ induced by the vertices of W, and for $X \subseteq E$ we write $G[X]$ for the graph (V, X) with vertex set V and edge set X.

DEFINITION 2.1. *Let $G = (V, E)$ be a graph and let t be a positive integer. A* (vertex) t-ranking*, called* ranking *for short if there is no ambiguity, is a coloring $\phi : V \to \{1, \ldots, t\}$ such that for every pair of vertices x and y with $\phi(x) = \phi(y)$ and for every path between x and y there is a vertex z on this path with $\phi(z) > \phi(x)$. The* vertex ranking number *of G, $\chi_r(G)$, is the smallest value t for which the graph G admits a t-ranking.*

By definition adjacent vertices have different colors in any t-ranking; thus any t-ranking is a proper t-coloring. Hence $\chi_r(G)$ is bounded below by the *chromatic number* $\chi(G)$. A vertex $\chi_r(G)$-ranking of G is said to be an *optimal (vertex) ranking* of G.

The edge ranking problem is closely related to the vertex ranking problem.

DEFINITION 2.2. *Let $G = (V, E)$ be a graph and let t be a positive integer. An* edge t-ranking *is an edge coloring $\phi' : E \to \{1, \ldots, t\}$ such that for every pair of edges e and f with $\phi'(e) = \phi'(f)$ and for every path between e and f there is an edge g on this path with $\phi'(g) > \phi'(e)$. The* edge ranking number $\chi'_r(G)$ *is the smallest value of t such that G has an edge t-ranking.*

REMARK 2.3. *There is a one-to-one correspondence between the edge t-rankings of a graph G and the vertex t-rankings of its line graph $L(G)$. Hence $\chi'_r(G) = \chi_r(L(G))$.*

An edge t-ranking of a graph G is a particular proper edge coloring of G. Hence, $\chi'_r(G)$ is bounded below by the *chromatic index* $\chi'(G)$. An edge $\chi'_r(G)$-ranking of G is said to be an *optimal edge ranking* of G.

As shown in [8], the vertex ranking number of a connected graph is equal to its minimum elimination tree height plus one. Thus, (vertex) separators and edge separators are a convenient tool for investigating rankings of graphs. A subset $S \subseteq V$ of a graph $G = (V, E)$ is said to be a *separator* if $G[V \setminus S]$ is disconnected. A subset $R \subseteq E$ of a graph $G = (V, E)$ is said to be an *edge separator* (or *edge cut*) if $G[E \setminus R]$ is disconnected.

In this paper we use the *separator tree* for studying vertex rankings. This concept is closely related to elimination trees (cf. [3, 8, 18]).

DEFINITION 2.4. *Given a vertex t-ranking $\phi : V \to \{1, 2, \ldots, t\}$ of a connected graph $G = (V, E)$, we assign a rooted tree $T(\phi)$ to it by an inductive construction such that a separator of a certain induced subgraph of G is assigned to each internal node of $T(\phi)$ and the vertices of each set assigned to a leaf of $T(\phi)$ have pairwise different colors.*

1. *If no color occurs more than once in G, then $T(\phi)$ consists of a single vertex r (called root) and the vertex set of G is assigned to r.*

2. *Otherwise, let i be the largest color assigned to more than one vertex by ϕ. Then $\{i+1, i+2, \ldots, t\}$ has to be a separator S of G. We create a root r of $T(\phi)$ and assign S to r. (The induced subgraph of G corresponding to the subtree of T rooted at r will be G itself.) Assuming that a separator tree $T_i(\phi)$ with root r_i has already been defined for each connected component G_i of the graph $G[V \setminus S]$, the children of r in $T(\phi)$ will be the vertices r_i and the subtree of $T(\phi)$ rooted at r_i will be $T_i(\phi)$.*

The rooted tree $T(\phi)$ is said to be a separator tree *of G.*

Notice that all vertices of G assigned to nodes of $T(\phi)$ on a path from a leaf to the root have different colors.

3. Unbounded ranking. It is still unknown whether the EDGE RANKING problem "Given a graph G and a positive integer t, decide whether $\chi'_r(G) \leq t$" is NP-complete. Clearly, by Remark 2.3 this problem is equivalent to the VERTEX RANKING problem "given a graph G and a positive integer t, decide whether $\chi_r(G) \leq t$" when restricted to line graphs.

On the other hand, it is a consequence of the NP-completeness of the minimum elimination tree height problem shown by Pothen in [20] and the equivalence of this problem with the VERTEX RANKING problem [6, 8] that the VERTEX RANKING problem is NP-complete even when restricted to graphs that are the complements of bipartite graphs, the so-called cobipartite graphs.

For reasons of self-containedness, we start with a short proof of the NP-completeness of VERTEX RANKING, when restricted to cobipartite graphs. The following problem, called BALANCED COMPLETE BIPARTITE SUBGRAPH (abbreviated BCBS) is NP-complete. This is Problem GT24 of [11].

INSTANCE: A bipartite graph $G = (V, E)$ and a positive integer k.
QUESTION: Are there two disjoint subsets $W_1, W_2 \subseteq V$ such that $|W_1| = |W_2| = k$ and such that $u \in W_1$, $v \in W_2$ implies that $\{u, v\} \in E$?

THEOREM 3.1. VERTEX RANKING *is NP-complete for cobipartite graphs.*

Proof. Clearly the problem is in NP. NP-hardness is shown by reduction from BCBS.

Let a bipartite graph $G = (V_1, V_2, E)$ and a positive integer k be given. Let $\overline{G}$ be the complement of G; thus $\overline{G}$ is a cobipartite graph.

We claim that G has a balanced complete bipartite subgraph with $2 \cdot k$ vertices if and only if $\overline{G}$ has an $(n-k)$-ranking.

Suppose we have sets $W_1 \subseteq V_1$, $W_2 \subseteq V_2$, such that $|W_1| = |W_2| = k$ and such that for all $u \in W_1$, $v \in W_2$: $\{u, v\} \in E$. We now construct an $(n-k)$-ranking of $\overline{G}$. Write $W_i = \{v_1^{(i)}, \ldots, v_k^{(i)}\}$ for $i \in \{1, 2\}$, and write $V \setminus (W_1 \cup W_2) = \{v'_1, \ldots, v'_{n-2\cdot k}\}$. We define a vertex ranking ϕ of $\overline{G}$ as follows:

$$\begin{aligned} \phi(v_j^{(1)}) = \phi(v_j^{(2)}) &= j \quad \text{for all } j, \qquad 1 \le j \le k. \\ \phi(v'_j) &= k + j \quad \text{for all } j, \qquad 1 \le j \le n - 2 \cdot k. \end{aligned}$$

One can easily observe that ϕ is a vertex $(n-k)$-ranking.

Next, let ϕ be an $(n-k)$-ranking for $\overline{G}$. Since $\overline{G}$ is a cobipartite graph, for each color, there can be at most two vertices with that color, one lying in V_1 and the other in V_2. Therefore, we have k pairs $v_j^{(1)}$ and $v_j^{(2)}$ with $\phi(v_j^{(1)}) = \phi(v_j^{(2)})$ and we can assume that $W_1 = \{v_j^{(1)} | 1 \le j \le k\} \subseteq V_1$ and $W_2 = \{v_j^{(2)} | 1 \le j \le k\} \subseteq V_2$.

Now we show that the subgraph induced by the set $W_1 \cup W_2$ forms a balanced complete bipartite subgraph in G. To show this, we prove that each pair of vertices $u \in W_1$, $v \in W_2$ is not adjacent in $\overline{G}$. Suppose $v_i^{(1)}$ and $v_j^{(2)}$ are adjacent in $\overline{G}$. Then, the colors of these vertices must be different. Furthermore, assume w.l.o.g. that $\phi(v_i^{(1)}) < \phi(v_j^{(2)})$. Then we have a path $(v_j^{(1)}, v_i^{(1)}, v_j^{(2)})$ with $\phi(v_i^{(1)}) < \phi(v_j^{(1)}) = \phi(v_j^{(2)})$ contradicting the fact that ϕ is a ranking. Hence, the subgraph induced by $W_1 \cup W_2$ is indeed a balanced complete bipartite subgraph. This proves the claim and the NP-completeness of VERTEX RANKING. □

We show that the analogous result holds for bipartite graphs as well.

THEOREM 3.2. VERTEX RANKING *remains NP-complete for bipartite graphs.*

Proof. We transform the VERTEX RANKING for arbitrary graphs without isolated vertices into that for bipartite graphs.

Given the graph G, we construct a graph $G' = (V', E')$. We define

$$V' = V \cup \{(e, i) \mid e \in E, 1 \le i \le t + 1\}$$

and

$$E' = \{\{v, (e, i)\} \mid v \in V, e \in E, 1 \le i \le t + 1 \text{ where } v \in e\}.$$

Clearly, the constructed graph G' is a bipartite graph. Now we show that G has a t-ranking if and only if G' has a $(t+1)$-ranking.

Suppose G has a t-ranking $\phi : V \to \{1, \ldots, t\}$. We construct a coloring $\hat{\phi}$ for G' in the following way. For the vertices $v \in V$ we set $\hat{\phi}(v) = \phi(v) + 1$ and for the vertices $(e, i) \in V' \setminus V$ we set $\hat{\phi}((e, i)) = 1$. Clearly $\hat{\phi}$ is a $(t+1)$-ranking of G'.

On the other hand, let $\hat{\phi} : V' \to \{1, \ldots, t+1\}$ be a $(t+1)$-ranking of G'. We show that $\hat{\phi}(v) > 1$ for every vertex $v \in V$. Suppose for a contradiction that v is a vertex of V with $\hat{\phi}(v) = 1$. Let $e = \{v, w\}$ be an edge incident to v in G. Hence, v is adjacent to $(e, 1), (e, 2), \ldots, (e, t+1)$ in G'. Then $\hat{\phi}(v) = 1$ implies $\hat{\phi}((e, i)) > 1$ for $i = 1, 2, \ldots, t+1$. Since $\hat{\phi}$ is a $(t+1)$-ranking, there are l, l' with $l \neq l'$ such that $\hat{\phi}((e, l)) = \hat{\phi}((e, l'))$, implying a path $((e, l), v, (e, l'))$ which contradicts the assumption that $\hat{\phi}$ is a ranking. This proves that $\hat{\phi}(v) > 1$ holds for every vertex $v \in V$. As a consequence, for each edge $e = \{u, v\} \in E$, there is a vertex $(e, i) \in V'$ with $\hat{\phi}((e, i)) < \min(\hat{\phi}(u), \hat{\phi}(v))$. Thus, changing $\hat{\phi}$ on $V' \setminus V$ to $\hat{\phi}((e, i)) = 1$ for all $(e, i) \in V'$, we obtain another $(t+1)$-ranking of G'. Now we define $\phi(v) = \hat{\phi}(v) - 1$ for every $v \in V$. The coloring ϕ is a t-ranking of G, since the existence of a path between two vertices v and w of G such that $\phi(v) = \phi(w)$ and all inner vertices have smaller colors implies the existence of a path from v to w in G' with $\hat{\phi}(v) = \hat{\phi}(w)$ and all inner vertices having smaller colors, contradicting the fact that $\hat{\phi}$ is a $(t+1)$-ranking of G'. □

4. Bounded ranking. We show that the "bounded" ranking problems—"Given a graph G, decide whether $\chi_r(G) \leq t$ ($\chi'_r(G) \leq t$)"—are solvable in linear time for any fixed t. This will be done by verifying that the corresponding graph classes are closed under certain operations.

DEFINITION 4.1. *An* edge contraction *is an operation of replacing two adjacent vertices u and v of a graph G by a vertex adjacent to all vertices that were adjacent to u or v. An* edge lift *is an operation of replacing two adjacent edges $\{u, w\}$ and $\{w, v\}$ of a graph G by one edge $\{u, v\}$.*

DEFINITION 4.2. *A graph H is a* minor *of the graph G if H can be obtained from G by a series of the following operations: vertex deletion, edge deletion, and edge contraction. A graph class $\mathcal{G}$ is* minor closed *if every minor H of every graph $G \in \mathcal{G}$ also belongs to $\mathcal{G}$.*

LEMMA 4.3. *The class of graphs satisfying $\chi_r(G) \leq t$ is minor closed for any fixed t.*

Proof. Since vertex/edge deletion cannot create new paths between monochromatic pairs of vertices, we only have to show that edge contraction does not increase the ranking number. Let $G = (V, E)$ be a graph with $\chi_r(G) \leq t$, and assume $H = (V', E')$ is obtained from G by contracting the edge $\{u, v\} \in E$ into a new vertex $\widehat{uv}$. Suppose ϕ is a t-ranking of G. We construct a coloring $\hat{\phi} : V' \to \{1, 2, \ldots, t\}$ of H as follows:

$$\hat{\phi}(x) = \begin{cases} \phi(x) & \text{if } x \in V \setminus \{u, v\} \\ \max(\phi(u), \phi(v)) & \text{if } x = \widehat{uv}. \end{cases}$$

Suppose $\hat{\phi}$ is not a t-ranking of H. Then there is a path $P = (x_0, x_1, \ldots, x_s)$, $s \geq 1$, of H such that $\hat{\phi}(x_0) = \hat{\phi}(x_s) > \hat{\phi}(x_i)$ for every $i \in \{1, 2, \ldots, s-1\}$. Since ϕ is a t-ranking of G the vertex $\widehat{uv}$ must occur in the path P. Depending on its neighbors in P we can "decontract" $\widehat{uv}$ in the path P into u, v, u—v or v—u getting a path P' of G violating the ranking condition, in contradiction to the choice of ϕ. □

COROLLARY 4.4. *For each fixed t, the class of graphs satisfying $\chi_r(G) \leq t$ can be recognized in linear time.*

Proof. In [1], using results from Robertson and Seymour [22, 23], it is shown that every minor closed class of graphs that does not contain all planar graphs has a linear time recognition algorithm. The result now follows directly from Lemma 4.3. □

Regarding edge rankings, a simple argument yields a much stronger assertion as follows.

THEOREM 4.5. *For each fixed t, the class of connected graphs satisfying $\chi'_r(G) \leq t$ can be recognized in constant time.*

Proof. For any fixed t, there are only a finite number of connected graphs G with $\chi'_r(G) \leq t$, as necessary conditions are that the maximum degree of G is at most t, and the diameter of G is bounded by $2^t - 1$. □

Certainly, the above theorem immediately implies that the graphs G satisfying $\chi'_r(G) \leq t$ can be recognized in linear time by inspecting the connected components separately. This result might have also been obtained via more involved methods using results of Robertson and Seymour on graph immersions [21]. Similarly, one can show that for fixed t and d, the class of connected graphs with $\chi_r(G) \leq t$ and maximum vertex degree d can be recognized in constant time.

DEFINITION 4.6. *A graph H is an* immersion *of the graph G if H can be obtained from G by a series of the following operations: vertex deletion, edge deletion, and edge lift. A graph class $\mathcal{G}$ is* immersion closed *if every immersion H of a graph $G \in \mathcal{G}$ also belongs to $\mathcal{G}$.*

The proof of the following lemma is similar to that of Lemma 4.3 and is omitted.

LEMMA 4.7. *The class of graphs satisfying $\chi'_r(G) \leq t$ is immersion closed for any fixed t.*

Linear-time recognizability of the class of graphs satisfying $\chi'_r(G) \leq t$ now also follows from Lemma 4.7, the results of Robertson and Seymour, and the fact that graphs with $\chi'_r(G) \leq t$ have treewidth at most $2t + 2$.

5. Computing the vertex ranking number on graphs with bounded treewidth. In this section, we show that one can compute $\chi_r(G)$ of a graph G with treewidth at most k in polynomial time, for any fixed k. Such a graph is also called a partial k-tree. This result implies polynomial-time computability of the vertex ranking number for any class of graphs with a uniform upper bound on the treewidth, e.g., outerplanar graphs, series-parallel graphs, Halin graphs.

The notion of treewidth has been introduced by Robertson and Seymour (see, e.g., [22]). See [2] for an overview on this notion.

DEFINITION 5.1. *A* tree-decomposition *of a graph $G = (V, E)$ is a pair (X, T) with $X = \{X_i \mid i \in I\}$ being a collection of subsets of V, and $T = (I, F)$ being a tree, such that*

(i) $\bigcup_{i \in I} X_i = V$;
(ii) *for all edges $\{v, w\} \in E$ there is an $i \in I$ with $v, w \in X_i$;*
(iii) *for all $i, j, k \in I$: if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.*

The width *of a tree-decomposition (X, T) with $\{X_i \mid i \in I\}$ is $\max_{i \in I} |X_i| - 1$. The* treewidth *of a graph $G = (V, E)$ is the minimum width over all tree-decompositions of G.*

When the treewidth of $G = (V, E)$ is bounded by a constant k, one can find in $O(n)$ time a tree-decomposition (X, T) of width at most k, such that $I = O(n)$ and T is a rooted binary tree [1]. Denote the root of T as r. We say (X, T) is a rooted binary tree-decomposition.

DEFINITION 5.2. *A* terminal graph *is a triple (V, E, Z), with (V, E) being an undirected graph, and $Z \subseteq V$ being a subset of the vertices, called the* terminals.

To each node i of a rooted binary tree-decomposition (X, T) of graph $G = (V, E)$, we associate the terminal graph $G_i = (V_i, E_i, X_i)$, where $V_i = \bigcup\{X_j \mid j = i \text{ or } j \text{ is a descendant of } i\}$, and $E_i = \{\{v, w\} \in E \mid v, w \in V_i\}$. As shorthand notation we write $p(v, w, G, \phi, \alpha)$, if and only if there is a path in G from v to w with all internal vertices having colors, smaller than α under coloring ϕ. If $p(v, w, G, \phi, \alpha)$, we denote with $\mathcal{P}(v, w, G, \phi, \alpha)$ the set of paths in G from v to w with all internal vertices having colors (using color function ϕ), smaller than α. In the following, suppose t is given.

DEFINITION 5.3. *Let $G = (V, E, Z)$ be a terminal graph, and let $\phi : V \to \{1, \ldots, t\}$ be a vertex t-ranking of (V, E). The* characteristic *of ϕ, $Y(\phi)$, is the quadruple $(\phi|_Z, f_1, f_2, f_3)$, where*

(i) *$\phi|_Z$ is the function ϕ, restricted to domain Z;*

(ii) *$f_1 : Z \times \{1, \ldots, t\} \to \{$true,false$\}$, is defined by: $f_1(v, i) =$ true if and only if $\phi(v) = i$ or there is a vertex $x \in V$ with $\phi(x) = i$ and $p(v, x, G, \phi, i)$;*

(iii) *$f_2 : Z \times Z \times \{1, \ldots, t\} \to \{$true,false$\}$, is defined by: $f_2(v, w, i) =$ true, if and only if there is a vertex $x \in V$ with $\phi(x) = i$, $p(v, x, G, \phi, i)$ and $p(w, x, G, \phi, i)$; and*

(iv) *$f_3 : Z \times Z \to \{1, \ldots, t, \infty\}$ is defined by: $f_3(v, w)$ is the smallest integer t' such that $p(v, w, G, \phi, t')$. If $\{v, w\} \in E$ then $f_3(v, w) = 0$, and if there is no path from v to w in G, then $f_3(v, w) = \infty$.*

DEFINITION 5.4. *A set of characteristics S of vertex t-rankings of a terminal graph G is* a full set of characteristics of vertex t-rankings for G *(in short, a full set for G), if and only if for every vertex t-ranking ϕ of G, $Y(\phi) \in S$.*

A set $\mathcal{C}$ of vertex t-rankings of a terminal graph G is an example set of vertex t-rankings for G *(in short, an example set for G), if and only if for every vertex t-ranking ϕ of G, there is a $\hat{\phi} \in \mathcal{C}$ with $Y(\phi) = Y(\hat{\phi})$, or, equivalently, the set of characteristics of the elements of $\mathcal{C}$ forms a full set of characteristics of vertex t-rankings for G.*

If $t = O(\log n)$, then a full set of characteristics of vertex t-rankings of $G = (V, E, Z)$ (with $|Z| \leq k+1$, k constant) has size polynomial in V: there are $O(log^{k+1} n)$ possible values for $\phi|_Z$, $2^{O((k+1)\log n)}$ possible values for f_1, $2^{O((k+1)^2 \log n)}$ possible values for f_2, and $O(\log^{\frac{1}{2}k(k+1)} n)$ possible values for f_3. The following lemma, given in [3], shows that we can ensure this property for graphs with treewidth at most k for fixed k.

LEMMA 5.5. *If the treewidth of $G = (V, E)$ is at most k, then $\chi_r(G) = O(k \cdot \log n)$.*

Let (X, T) be a rooted binary tree-decomposition of G. Suppose $j \in I$ is a descendant of $i \in I$ in T. Suppose ϕ is a vertex t-ranking of G_i. The restriction of ϕ to G_j is the function $\phi|_{G_j} : V_j \to \{1, \ldots, t\}$, defined by $\forall v \in V_j : \phi|_{G_j}(v) = \phi(v)$. Clearly, $\phi|_{G_j}$ is a vertex t-ranking of G_j. If $\hat{\phi}$ is another vertex t-ranking of G_j, we define the function $R(\phi, \hat{\phi}) : V_i \to \{1, \ldots, t\}$, by

$$R(\phi, \hat{\phi})(v) = \begin{cases} \phi(v) & \text{if } v \in V_i \setminus V_j, \\ \hat{\phi}(v) & \text{if } v \in V_j. \end{cases}$$

LEMMA 5.6. *Let (X, T) be a rooted binary tree-decomposition of $G = (V, E)$. Let j be a descendant of i. Let ϕ be a vertex t-ranking of G_i and $\hat{\phi}$ a vertex t-ranking of G_j. If $Y(\phi|_{G_j}) = Y(\hat{\phi})$, then $R(\phi, \hat{\phi})$ is a vertex t-ranking of G_i, and $Y(\phi) = Y(R(\phi, \hat{\phi}))$.*

Proof. For brevity, we write $\tilde{\phi} = R(\phi, \hat{\phi})$, $W_1 = (V_i \setminus V_j) \cup X_j$, $W_2 = V_j \setminus X_j$, and $Y(\phi|_{G_j}) = Y(\hat{\phi}) = (\hat{\phi}|_{X_j}, f_1, f_2, f_3)$.

We start by proving two claims.

CLAIM 5.6.1. *For all $v, w \in W_1$ and all $t' \leq t$, $p(v, w, G_i, \phi, t') \Leftrightarrow p(v, w, G_i, \tilde{\phi}, t')$.*

Proof. Let $v, w \in W_1$, and suppose we have a path $P \in \mathcal{P}(v, w, G_i, \phi, t')$. We consider those parts of the path P that are part of G_j: write $P = (P_0, P_0', P_1, P_1', \ldots, P_{r-1}, P_{r-1}', P_r)$, such that each P_α $(0 \leq \alpha \leq r)$ is a path with all vertices in W_1, and each P_α' $(0 \leq \alpha \leq r-1)$ is a path in G_j. (Each path is a collection of successive edges, i.e., the last vertex of a path is the first vertex of the next path.) Write v_α for the first vertex on path P_α' and w_α for the last vertex on path P_α' $(0 \leq \alpha \leq r-1)$. Note that $P_\alpha' \in \mathcal{P}(v_\alpha, w_\alpha, G_j, \phi, t')$; hence $f_3(v_\alpha, w_\alpha) \leq t'$. We now have that there also exists a path $P_\alpha'' \in \mathcal{P}(v_\alpha, w_\alpha, G_j, \hat{\phi}, t')$. (In other words, there exists a path from v_α to w_α in G_j such that all colors of internal vertices are smaller than t', using coloring ϕ (or, equivalently, $\phi|_{G_j}$). As $\phi|_{G_j}$ and $\hat{\phi}$ have the same characteristics, there also exists such a path using color function $\hat{\phi}$.) Now, the path formed by the sequence $(P_0, P_0'', P_1, P_1'', \ldots, P_{r-1}, P_{r-1}'', P_r)$ is a path in G_i between v and w with all colors of internal vertices smaller than t'; hence $p(v, w, G_i, \tilde{\phi}, t')$. This shows: $p(v, w, G_i, \phi, t') \Rightarrow p(v, w, G_i, \tilde{\phi}, t')$. $p(v, w, G_i, \phi, t') \Leftarrow p(v, w, G_i, \tilde{\phi}, t')$ can be shown in the same way. ∎

CLAIM 5.6.2. *For all $v \in W_1$ and all $t' \leq t$, there exists a vertex $w \in V_i$ $(w \in V_j)$ with $p(v, w, G_i, \phi, t')$ and $\phi(w) = t'$, if and only if there exists a vertex $w' \in V_i$ $(w' \in V_j)$ with $p(v, w', G_i, \tilde{\phi}, t')$, and $\tilde{\phi}(w') = t'$.*

Proof. Let $w \in V_i$ with $p(v, w, G_i, \phi, t')$ and $\phi(w) = t'$. If $w \in W_1$, then, by Claim 5.6.1, we have $p(v, w, G_i, \tilde{\phi}, t')$. Otherwise, let x be the last vertex on a path $P \in \mathcal{P}(v, w, G_i, \phi, t')$ that belongs to W_1. Write $P = (P', P'')$, where x is the last vertex of P' and the first vertex of P''. $P' \in \mathcal{P}(v, x, G_i, \phi, t')$; hence there exists a path $Q' \in \mathcal{P}(v, x, G_i, \tilde{\phi}, t')$. $P'' \in \mathcal{P}(x, w, G_j, \phi|_{G_j}, t')$; hence $f_1(x, t') = \text{true}$. Using equality of the characteristics of $\phi|_{G_j}$ and $\hat{\phi}$, we have that there exists a vertex $w' \in V_j$ with $\hat{\phi}(w') = t' = \tilde{\phi}(w')$ and a path $Q'' \in \mathcal{P}(x, w', G_j, \hat{\phi}, t')$. Furthermore, $\tilde{\phi}(x) < t'$: $x \in W_1$ is adjacent to a vertex in W_2, thus $x \in X_j$, and $\phi|_{X_j} = \hat{\phi}|_{X_j} = \tilde{\phi}|_{X_j}$, so $\tilde{\phi}(x) = \phi(x) < t'$. (The latter inequality holds because x is an internal vertex of $P \in \mathcal{P}(v, w, G_i, \phi, t')$.) Therefore, (Q', Q'') is a path from v to w' in G_i with all internal vertices of color (under color function $\tilde{\phi}$) smaller than t', and hence $p(v, w', G_i, \tilde{\phi}, t')$. The reverse implication of the claim can be shown in a similar way. ∎

We now show that $\tilde{\phi}$ is a vertex t-ranking, or, equivalently, that for all $v, w \in V_i$, if $\tilde{\phi}(v) = \tilde{\phi}(w)$, then $\neg p(v, w, G_i, \tilde{\phi}, \tilde{\phi}(v))$. Let $v, w \in V_i$ with $\tilde{\phi}(v) = \tilde{\phi}(w) = t'$, $v \neq w$ be given. We consider four cases.

1. $v, w \in W_1$. If $p(v, w, G_i, \tilde{\phi}, t')$, then by Claim 5.6.1, $p(v, w, G_i, \phi, t')$, and $\phi(v) = \tilde{\phi}(v) = t'$, $\phi(w) = \tilde{\phi}(w) = t'$; hence ϕ is not a vertex ranking, which is a contradiction.

2. $v \in W_1$, $w \in W_2$. If $p(v, w, G_i, \tilde{\phi}, t')$, then by Claim 5.6.2, there exists a $w' \in V_i$ with $p(v, w', G_i, \phi, t')$ and $\phi(w') = \phi(v)$; hence, again ϕ is not a vertex ranking, which is a contradiction.

3. $w \in W_1$, $v \in W_2$. This is similar to Case 2.

4. $v, w \in W_2$. Let $P \in \mathcal{P}(v, w, G_i, \tilde{\phi}, t')$. If all vertices on P belong to W_2, then P is a path in G_j, and hence $\hat{\phi}$ was not a vertex ranking of G_j, which is a contradiction. So, there exist vertices on P that belong to W_1.

Let x be the first vertex on P that belongs to W_1. Then $\tilde{\phi}(x) < t'$ and $P = (P_1, P_2)$, with $P_1 \in \mathcal{P}(v, x, G_j, \tilde{\phi}, t')$ and $P_2 \in \mathcal{P}(x, w, G_i, \tilde{\phi}, t')$. By Claim 5.6.2, there must exist vertices $v', w' \in V_j$ with $\phi(v') = \phi(w') = t'$ and paths Q_1, Q_2, with $Q_1 \in \mathcal{P}(v', x, G_j, \phi, t')$, $Q_2 \in \mathcal{P}(x, w', G_i, \phi, t')$. Since x is a vertex in W_1 that is adjacent to a vertex in W_2, we have that $x \in X_j$, and $\phi|_{X_j} = \hat{\phi}|_{X_j} = \tilde{\phi}|_{X_j}$, $\phi(x) = \tilde{\phi}(x) < t'$. Therefore, the path $Q = (Q_1, Q_2)$ is a path from v' to w' with all internal vertices of color (with color function ϕ) less than t'. Hence ϕ is not a vertex ranking, which is a contradiction.

It remains to show that $Y(\phi) = Y(\tilde{\phi})$. Clearly, $\phi|_{X_i} = \tilde{\phi}|_{X_i}$. Suppose $Y(\phi) = (\phi|_{X_i}, g_1, g_2, g_3)$ and $Y(\tilde{\phi}) = (\phi|_{X_i}, g_1', g_2', g_3')$. It follows directly from Claim 5.6.2 that $g_1 = g_1'$.

Consider $v, w \in X_i$, $t' \in \{1, \ldots, t\}$. Suppose $g_2(v, w, t') = \text{true}$. Let $x \in V_i$ be the vertex with $\phi(x) = t'$ and $p(v, x, G_i, \phi, t')$ and $p(w, x, G_i, \phi, t')$. If $x \in W_1$, then by Claim 5.6.1, $p(v, x, G_i, \tilde{\phi}, t')$ and $p(w, x, G_i, \tilde{\phi}, t')$; hence $g_2'(v, w, t') = \text{true}$. If $x \in W_2$, then let $P_1 \in \mathcal{P}(v, x, G_i, \phi, t')$ and let $P_2 \in \mathcal{P}(w, x, G_i, \phi, t')$. We can write $P_1 = (P_{11}, P_{12})$ with $P_{11} \in \mathcal{P}(v, y, G_i, \phi, t')$, $P_{12} \in \mathcal{P}(y, x, G_j, \phi, t')$, and $y \in X_j$. (Let y be the last vertex in X_j on P_1.) Note that $\phi(y) = \tilde{\phi}(y) < t'$, as $\phi|_{X_j} = \tilde{\phi}|_{X_j} = \hat{\phi}|_{X_j}$. Similarly, we can write $P_2 = (P_{21}, P_{22})$ with $P_{21} \in \mathcal{P}(w, z, G_i, \phi, t')$, $P_{22} \in \mathcal{P}(z, x, G_j, \phi, t')$, and $z \in X_j$. This implies that $f_2(y, z, t')$ is true. Hence, there is a vertex x' with paths $P_{12}' \in \mathcal{P}(y, x', G_j, \tilde{\phi}, t')$ and $P_{22}' \in \mathcal{P}(z, x', G_j, \tilde{\phi}, t')$, and with $\tilde{\phi}(x') = t'$. Also, by Claim 5.6.1 we have paths $P_{11}' \in \mathcal{P}(v, y, G_i, \tilde{\phi}, t')$ and $P_{21}' \in \mathcal{P}(w, z, G_i, \tilde{\phi}, t')$. Now, using path (P_{11}', P_{12}') from v to x' and path (P_{21}', P_{22}') from w to x', it follows that $g_2'(v, w, t')$ is true. So $g_2(v, w, t') \Rightarrow g_2'(v, w, t')$. An almost identical argument shows $g_2'(v, w, t') \Rightarrow g_2(v, w, t')$; hence $g_2 = g_2'$.

Finally, it follows directly from Claim 5.6.1 that $g_3 = g_3'$. □

We now describe our algorithm. After a rooted binary tree-decomposition (X, T) of $G = (V, E)$ has been found (in linear time [1]), the algorithm computes a full set and an example set for every node $i \in I$, in a bottom-up order. Clearly, when we have a full set for the root node of T, we can determine whether G has a vertex t-ranking, as we only have to check whether the full set of the root is nonempty. If so, any element of the example set of the root node gives us a vertex t-ranking of G.

It remains to show that we can compute for any node $i \in I$ a full set and an example set, given a full set and an example set for each of the children of $i \in I$. This is straightforward for the case that i is a leaf node: enumerate all functions $\phi : X_i \to \{1, \ldots, t\}$; for each such function ϕ, test whether it is a vertex t-ranking of G_i, and if so, put ϕ in the example set, and $Y(\phi)$ in the full set of characteristics.

Next, suppose $i \in I$ has two children j_1 and j_2. (If i has one child j_1, then we can add another child j_2, which is a leaf in T and has $X_{j_2} = X_i$.) Suppose we have example sets $\mathcal{C}_1$, $\mathcal{C}_2$ for G_{j_1} and G_{j_2}. We compute a full set S and an example set $\mathcal{C}$ for G_i in the following way.

Initially, we take S and $\mathcal{C}$ to be empty.

For each triple (ϕ_1, ϕ_2, ϕ_3), where ϕ_1 is an element of $\mathcal{C}_1$, ϕ_2 is an element of $\mathcal{C}_2$, and ϕ_3 is an arbitrary function $\phi_3 : X_i \setminus (X_{j_1} \cup X_{j_2}) \to \{1, \ldots, t\}$, do the following:

(i) Check whether for all $v \in X_{j_1} \cap X_{j_2}$, $\phi_1(v) = \phi_2(v)$. If not, skip the following steps and proceed with the next triple.

(ii) Compute the function $\phi : X_i \to \{1, \ldots, t\}$, defined as follows:

$$\phi(v) = \begin{cases} \phi_1(v) & \text{if } v \in V_{j_1} \\ \phi_2(v) & \text{if } v \in V_{j_2} \\ \phi_3(v) & \text{if } v \in X_i \setminus (X_{j_1} \cup X_{j_2}). \end{cases}$$

(iii) Check whether ϕ is a vertex t-ranking of G_i. If not, skip the following steps and proceed with the next triple.

(iv) Compute $Y(\phi)$.

(v) If $Y(\phi) \notin S$, then put $Y(\phi)$ in S and put ϕ in $\mathcal{C}$.

We claim that the resulting sets S and $\mathcal{C}$ form a full set and an example set for G_i. Consider an arbitrary vertex t-ranking $\hat{\phi}$ of G_i. Let $\phi_1 \in \mathcal{C}_1$ be the vertex t-ranking of G_{j_1} that has the same characteristic as $\hat{\phi}|_{G_{j_1}}$. By definition of example set, ϕ_1 must exist. Similarly, let $\phi_2 \in \mathcal{C}_2$ fulfill $Y(\phi_2) = Y(\hat{\phi}|_{G_{j_2}})$. Let $\phi_3 : X_i \setminus (X_{j_1} \cup X_{j_2}) \to \{1, \ldots, t\}$ be defined by $\phi_3(v) = \hat{\phi}(v)$ for all $v \in X_i \setminus (X_{j_1} \cup X_{j_2})$. When the algorithm processes the triple (ϕ_1, ϕ_2, ϕ_3), the first test will hold. Suppose ϕ is the function, computed in the second test. Now note that $\phi = R(R(\hat{\phi}, \phi_1), \phi_2)$. Hence, by Lemma 5.6, ϕ is a vertex t-ranking and has the same characteristic as $\hat{\phi}$. Hence, S will contain $Y(\phi)$, and $\mathcal{C}$ will contain a vertex t-ranking of G_i with the same characteristic as ϕ and $\hat{\phi}$.

As the size of a full set, and hence of an example set for graphs G_i, $i \in I$, is polynomial, it follows that the computation of a full set, and an example set from these sets associated with the children of the node, can be done in polynomial time. (There are a polynomial number of triples (ϕ_1, ϕ_2, ϕ_3). For each triple, the computation given above costs polynomial time.) As there are a linear number of nodes of the tree-decomposition, computing whether there exists a vertex t-ranking costs polynomial time (assuming $t = O(\log n)$). By testing for each applicable value of t (see Lemma 5.5) for the existence of vertex t-rankings of G, we obtain the following result.

THEOREM 5.7. *For any fixed k, there exists a polynomial time algorithm that determines the vertex ranking number of graphs G with treewidth at most k and finds an optimal vertex ranking of G.*

6. The equality $\chi_r = \chi$. In this section we consider questions related to the equality of the chromatic number and the vertex ranking number of graphs.

THEOREM 6.1. *If $\chi_r(G) = \chi(G)$ holds for a graph G, then G also satisfies $\chi(G) = \omega(G)$.*

Proof. Suppose that $G = (V, E)$ has a vertex t-ranking $\phi : V \to \{1, 2, \ldots, t\}$ with $t = \chi(G)$. We are going to consider the separator tree $T(\phi)$ of this t-ranking. Recall that $T(\phi)$ is a rooted tree and that to every internal node of $T(\phi)$ a subset of the vertex set of G is assigned such that the subset is a separator of the corresponding subgraph of G; namely, more than one component arises when all subsets on the path from the node to the root are deleted from the graph. Furthermore, all vertices assigned to the nodes of a path from a leaf to the root of $T(\phi)$ have pairwise different colors.

The goal of the following recoloring procedure is to show that either $\chi(G) = \omega(G)$ or we can recolor G to obtain a proper coloring with a smaller number of colors. However, the latter contradicts the choice of the $\chi(G)$-ranking ϕ.

We label the nodes of the tree $T(\phi)$ according to the following marking rules.

1. Mark a node s of $T(\phi)$ if the union $U(s)$ of all vertex sets assigned to all nodes on the path from s to the root is *not* a clique in G.

2. Also, mark a leaf l of $T(\phi)$ if the union $U(l)$ of all vertex sets assigned to all nodes on the path from l to the root is a clique in G, but $|U(l)| < t$.

Case 1. There is an unmarked leaf l.

We have $|U(l)| = t$ and $U(l)$ is a clique. Hence, $\omega(G) = \chi(G)$.

Case 2. There is no unmarked leaf.

We will show that this would enable us to recolor G saving one color, contradicting the choice of ϕ.

Since every leaf of $T(\phi)$ is marked, every path from a leaf to the root consists of marked nodes eventually followed by unmarked nodes. Consequently, there is a collection of marked branches of $T(\phi)$, i.e., subtrees of $T(\phi)$ induced by one node and all its descendants for which all nodes are marked and the father of the highest node of each branch is unmarked or the highest node is the root of $T(\phi)$ itself.

If the root of $T(\phi)$ is marked then we have exactly one marked branch, namely $T(\phi)$ itself. Then, by definition, the separator S assigned to the root is *not* a clique. However, none of its colors is used by the ranking for vertices in $V \setminus S$. Simply, any coloring of the separator S with fewer than $|S|$ colors will produce a coloring of G with fewer than $\chi(G)$ colors; this is a contradiction.

If the root is unmarked, then we have to work with a collection of b marked branches, $b > 1$. Notice that all color-1 vertices of G are assigned to leaves of $T(\phi)$ and that any leaf of $T(\phi)$ belongs to some marked branch B. We are going to recolor the graph G by recoloring the marked branches one by one such that the new coloring of G does not use color 1. Let us consider a marked branch B. Let h be its highest node in $T(\phi)$ and $S(h)$ the set assigned to h. Since h is marked but the root is unmarked, there must exist a vertex x of $S(h)$ and a vertex y belonging to $U(h)$ which are nonadjacent. Then $\phi(x) \neq \phi(y)$, since all vertices of $U(h)$ have pairwise different colors.

Assume $\phi(x) = 1$ or $\phi(y) = 1$. Then h is a leaf of $T(\phi)$. Hence, x and y, respectively, are the only color-1 vertices of G assigned to a node of B. We simply recolor x and y with $\max(\phi(x), \phi(y))$.

Finally, consider the case $\phi(x) \neq 1$ and $\phi(y) \neq 1$. All color-1 vertices in the subgraph of G corresponding to B are recolored with $\phi(x)$ and x is recolored with $\phi(y)$. By the construction of $T(\phi)$, this does not influence other parts of the graph, since they are separated by vertex sets with higher colors.

Having done this operation in every marked branch, eventually we get a new color assignment of G which is still a proper coloring (though usually not a ranking). Since all leaves of $T(\phi)$ are marked and no internal node of $T(\phi)$ contains color-1 vertices, color 1 is eliminated from G, contradicting the assumption $\chi_r(G) = \chi(G)$. Consequently, Case 2 cannot occur, implying $\chi(G) = \omega(G)$. This completes the proof. □

Clearly, $\chi_r(G) = \chi(G)$ does not imply that G is a perfect graph. (Trivial counterexamples are of the form $G = G' \cup K_{\chi_r(G')}$ where G' is an arbitrary imperfect graph.) On the other hand, if we require the equality on all induced subgraphs, then we remain with a relatively small class of graphs that is also called "trivially perfect" in the literature (cf. [12]).

THEOREM 6.2. *A graph $G = (V, E)$ satisfies $\chi_r(G[A]) = \chi(G[A])$ for every $A \subseteq V$ if and only if neither path P_4 nor cycle C_4 is an induced subgraph of G.*

Proof. The condition is necessary since $\chi_r(P_4) = \chi_r(C_4) = 3$ and $\chi(P_4) = \chi(C_4) = 2$.

Now let G be a P_4-free and C_4-free graph. The graphs with no induced P_4

and C_4 are precisely those in which every connected induced subgraph H contains a dominating vertex w, i.e., w is adjacent to all vertices of H [27]. Hence, the following efficient algorithm produces an optimal ranking in such graphs: if $H = (V', E')$ is connected, then we assign the color $\omega(H)$ to a dominating vertex w. Clearly, $\chi(H[V' \setminus \{w\}]) = \omega(H[V' \setminus \{w\}]) = \omega(H) - 1$, and it is easily seen that $\chi_r(H[V' \setminus \{w\}]) = \chi_r(H) - 1$ also holds; thus, induction can be applied. On the other hand, if H is disconnected, then an optimal ranking can be generated in each of its components separately. □

7. Edge rankings of complete graphs. While obviously $\chi_r(K_n) = n$, it is not easy to give a closed formula for the edge ranking number of the complete graph. The most convenient way to determine $\chi'_r(K_n)$ seems to introduce a function $g(n)$ by the rules

$$g(1) = -1,$$
$$g(2n) = g(n),$$
$$g(2n+1) = g(n+1) + n.$$

In terms of this $g(n)$, the following statement can be proved.

THEOREM 7.1. *For every positive integer n,*

$$\chi'_r(K_n) = \frac{n^2 + g(n)}{3}.$$

Proof. The assertion is obviously true for $n = 1, 2, 3$. For larger values of n we are going to apply induction.

Similarly to vertex t-rankings, the following property holds for every edge t-ranking of a graph $G = (V, E)$: if i is the largest color occurring more than once, then the edges with colors $i+1, i+2, \ldots, t$ form an edge separator of G. Moreover, doing an appropriate relabeling of these colors $i+1, i+2, \ldots, t$ we get a new edge t-ranking of G with the property that there is a color $j > i$ such that all edges with colors $j, j+1, \ldots, t$ form an edge separator of G which is minimal under inclusion.

We have to show that the best way to choose this edge separator R with respect to an edge ranking in a complete graph is by making the two components of $G[E \setminus R]$ as equal-sized as possible. Let us consider a K_n, $n \geq 4$. Let n_1 and n_2 be the numbers of vertices in the components; hence, $n_1 + n_2 = n$ and the corresponding edge separator has size $n_1 n_2$. Every edge ranking starting with this separator has at least

$$n_1 n_2 + \max\{\chi'_r(K_{n_1}), \chi'_r(K_{n_2})\} = n_1 n_2 + \chi'_r(K_{\max\{n_1, n_2\}})$$

colors, and there is indeed one using exactly that many colors. Defining $a_1 := \min(n_1, n_2)$ and repeating the same argument for $n' := n - a_1$, and so on, we eventually get a sequence of positive integers $a_1, \ldots, a_s$, for some s, such that $\sum_{i=1}^{s} a_i = n$ and

$$a_i \leq \sum_{i<j\leq s} a_j \quad \text{for all } i, \qquad 1 \leq i < s. \tag{1}$$

Notice that at least the last two terms of any such sequence are equal to 1. It is easy to see that the number of colors of any edge ranking represented by $a_1, \ldots, a_s$ is equal to $\sum_{1\leq i<j\leq s} a_i a_j$, consequently,

$$\chi'_r(K_n) = \min \sum_{1\leq i<j\leq s} a_i a_j = \binom{n}{2} - \max \sum_{i=1}^{s} \binom{a_i}{2},$$

subject to condition (1). Since a decreasing sort of the sequence maintains (1) we may assume $a_1 \geq a_2 \geq \cdots \geq a_s$. Thus, for each value of n, $\min \sum_{1 \leq i < j \leq s} a_i a_j$ is attained precisely by the unique sequence satisfying $a_i = \lfloor \frac{1}{2} \sum_{i \leq j \leq s} a_j \rfloor$ for all i, $1 \leq i < s$. In particular, we obtain

$$\chi'_r(K_n) = \chi'_r(K_{\lceil n/2 \rceil}) + \lfloor n/2 \rfloor \lceil n/2 \rceil.$$

Applying this recursion, it is not difficult to verify that, indeed, $\chi'_r(K_n)$ can be written in the form $\frac{1}{3}(n^2 + g(n))$, where $g(n)$ is the function defined above. □

Observing that $g(2^n) = -1$ for all $n \geq 1$, we obtain the following interesting result.

COROLLARY 7.2.

$$\chi'_r(K_{2^n}) = \frac{4^n - 1}{3}.$$

8. Conclusions. We studied algorithmic and graph theoretic properties of rankings of graphs. For many special classes of graphs, the algorithmic complexity of VERTEX RANKING is now known. However, the algorithmic complexity of VERTEX RANKING when restricted to chordal graphs or circle graphs is still unknown. Furthermore, it is not even known whether the EDGE RANKING problem is NP-complete.

We started a graph theoretic study of vertex ranking and edge ranking as a particular kind of proper (vertex) coloring and proper edge coloring, respectively. Much research has to be done in this direction. It is of particular interest which of the well-known problems in the theory of vertex colorings and edge colorings are also worth studying for vertex rankings and edge rankings.

REFERENCES

[1] H. L. BODLAENDER, *A linear time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[2] H. L. BODLAENDER, *A tourist guide through treewidth*, Acta Cybernetica, 11 (1993), pp. 1–23.

[3] H. L. BODLAENDER, J. R. GILBERT, H. HAFSTEINSSON, AND T. KLOKS, *Approximating treewidth, pathwidth and minimum elimination tree height*, J. Algorithms, 18 (1995), pp. 238–255.

[4] J. A. BONDY AND U. S. R. MURTY, *Graph Theory with Applications*, American Elsevier, New York, 1976.

[5] A. BRANDSTÄDT, *Special Graph Classes — A Survey*, Technical Report SM-DU-199, Schriftenreihe des FB Mathematik, Universität Duisburg, 1993.

[6] P. DE LA TORRE, R. GREENLAW, AND A. A. SCHÄFFER, *Optimal edge ranking of trees in polynomial time*, Algorithmica, 13 (1995), pp. 592–618.

[7] P. DE LA TORRE, R. GREENLAW, AND A. A. SCHÄFFER, *A Note on Deogun and Peng's Edge Ranking Algorithm*, Technical Report 93-13, Dept. of Computer Science, Univ. of New Hampshire, Durham, NH, 1993.

[8] J. S. DEOGUN, T. KLOKS, D. KRATSCH, AND H. MÜLLER, *On vertex ranking for permutation and other graphs*, in Proc. 11th Annual Symposium on Theoretical Aspects of Computer Science, P. Enjalbert, E. W. Mayr, and K. W. Wagner, eds., Lecture Notes in Computer Science 775, Springer-Verlag, Berlin, 1994, pp. 747–758.

[9] J. S. DEOGUN AND Y. PENG, *Edge ranking of trees*, Congr. Numer., 79 (1990), pp. 19–28.

[10] I. S. DUFF AND J. K. REID, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[11] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, New York, 1979.

[12] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[13] A. V. Iyer, H. D. Ratliff, and G. Vijayan, *Optimal node ranking of trees*, Inform. Processing Lett., 28 (1988), pp. 225–229.
[14] A. V. Iyer, H. D. Ratliff, and G. Vijayan, *On an edge ranking problem of trees and graphs*, Discrete Appl. Math., 30 (1991), pp. 43–52.
[15] M. Katchalski, W. McCuaig, and S. Seager, *Ordered colourings*, Discrete Math., 142 (1995), pp. 141–154.
[16] T. Kloks, *Treewidth–Computations and Approximations*, Lecture Notes in Computer Science 842, Springer-Verlag, New York, 1994.
[17] C. E. Leiserson, *Area efficient graph layouts for VLSI*, in Proc. 21st Annual IEEE Symposium on Foundations of Computer Science, 1980, IEEE Computer Society Press, Los Alamitos, CA, pp. 270–281.
[18] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
[19] J. Nevins and D. Whitney, eds., *Concurrent Design of Products and Processes*, McGraw-Hill, New York, 1989.
[20] A. Pothen, *The Complexity of Optimal Elimination Trees*, Technical Report CS-88-13, Pennsylvania State University, University Park, PA, 1988.
[21] N. Robertson and P. D. Seymour, *Graph minors. IV. Tree-width and well-quasi-ordering*, J. Combin. Theory Ser. B, 48 (1990), pp. 227–254.
[22] N. Robertson and P. D. Seymour, *Graph minors. V. Excluding a planar graph*, J. Combin. Theory Ser. B, 41 (1986), pp. 92–114.
[23] N. Robertson and P. D. Seymour, *Graph minors. XIII. The disjoint paths problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.
[24] A. A. Schäffer, *Optimal node ranking of trees in linear time*, Inform. Processing Lett., 33 (1989/90), pp. 91–96.
[25] P. Scheffler, *Node ranking and searching on graphs*, in Proc. 3rd Twente Workshop on Graphs and Combinatorial Optimization, Memorandum No. 1132, U. Faigle and C. Hoede, eds., Faculty of Applied Mathematics, University of Twente, the Netherlands, 1993, abstract.
[26] A. Sen, H. Deng, and S. Guha, *On a graph partition problem with application to VLSI layout*, Inform. Process. Lett., 43 (1992), pp. 87–94.
[27] E. S. Wolk, *The comparability graph of a tree*, Proc. AMS, 3 (1962), pp. 789–795.
[28] X. Zhou, M. A. Kashem, and T. Nishizeki, *Generalized edge-rankings of trees*, in Proc. 22nd Internat. Workshop Graph-Theoretic Concepts in Computer Science (WG'96), Lecture Notes in Computer Science 1197, F. d'Amore, P.G. Franciosa, and A. Marchetti-Spaccamela, eds., Springer-Verlag, Berlin, 1997, pp. 390–404.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 183–195, May 1998

AN OPTIMAL ACCEPTANCE POLICY FOR AN URN SCHEME*

ROBERT W. CHEN†, ALAN ZAME†, ANDREW M. ODLYZKO‡, AND LARRY A. SHEPP‡

Abstract. An urn contains m balls of value -1 and p balls of value $+1$. At each turn a ball is drawn randomly, without replacement, and the player decides before the draw whether or not to accept the ball, i.e., the bet where the payoff is the value of the ball. The process continues until all $m+p$ balls are drawn. Let $\overline{V}(m,p)$ denote the value of this acceptance (m,p) urn problem under an optimal acceptance policy. In this paper, we first derive an exact closed form for $\overline{V}(m,p)$ and then study its properties and asymptotic behavior. We also compare this acceptance (m,p) urn problem with the original (m,p) urn problem which was introduced by Shepp [*Ann. Math. Statist.*, 40 (1969), pp. 993–1010]. Finally, we briefly discuss some applications of this acceptance (m,p) urn problem and introduce a Bayesian approach to this optimal stopping problem. Some numerical illustrations are also provided.

Key words. optimal stopping, acceptance policy, urn models, Bayesian approach

AMS subject classifications. primary, 60G40; secondary, 60K99

PII. S0895480195282148

1. Introduction.

In [7], Shepp considered the following optimal **stopping** problem: An (m,p) urn contains m balls of value -1 and p balls of value $+1$, and the player is allowed to draw balls randomly, without replacement, until he wants to stop. Shepp was interested in finding, for what m and p, if there is an optimal drawing policy for which $V(m,p)$ is positive, where $V(m,p)$ is the value of this (m,p) urn problem under an optimal drawing policy. In particular, he showed that for every positive integer p there is a positive integer $\beta(p)$ for which $V(m,p) > 0$ or $= 0$, with $0 \le m \le \beta(p)$ or $m > \beta(p)$ accordingly. In [2, 3], Boyce, motivated by applications to financial and marketing problems, also studied this (m,p) urn problem. In [4], Chen and Hwang derived some new properties of $V(m,p)$ that give additional insight into the structure of the optimal drawing policy for this (m,p) urn problem.

In this paper, we study a new (m,p) urn problem that we call an **acceptance** (m,p) urn problem and that can be simply described as follows: An urn contains m balls of value -1 and p balls of value $+1$. At each turn a ball is drawn randomly, without replacement, and the player decides before the draw whether or not to **accept** the ball, i.e., the bet where the payoff is the value of the ball. The process will continue until all $m+p$ balls are drawn. We are interested in the value $\overline{V}(m,p)$ of this acceptance (m,p) urn problem under an optimal **acceptance** policy. We first derive an exact closed form for $\overline{V}(m,p)$ by a simple probabilistic argument and obtain inequalities of the form $\overline{V}(m,p) < \overline{V}(m+1,p+1)$, in the spirit of [3] and [4] for the original urn problem. Then we study the asymptotic behavior of $\overline{V}(m,p)$. We also compare this acceptance (m,p) urn problem with the original (m,p) urn problem. Finally, we briefly indicate an application of this acceptance urn version of the optimal policy problematics to (in-and-out) bond trading and introduce a Bayesian approach to this optimal stopping problem. Some numerical illustrations are also provided.

*Received by the editors February 27, 1995; accepted for publication (in revised form) January 30, 1997.

http://www.siam.org/journals/sidma/11-2/28214.html

†Department of Mathematics and Computer Science, University of Miami, Coral Gables, FL 33124 (chen@cs.miami.edu, zame@cs.miami.edu).

‡AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974 (amo@research.att.com, las@research.att.com).

2. Exact solutions of $\overline{V}(m,p)$. For each nonnegative integer m and p such that $m+p \geq 1$, let $A(m,p)$ be the expected value of accepting the current drawn ball from the (m,p) urn, assuming an optimal acceptance policy is followed after the current draw, and let $N(m,p)$ be the expected value of not accepting the current drawn ball from the (m,p) urn, assuming an optimal acceptance policy is followed after the current draw. It is clear that $\overline{V}(m,p) = \max\{A(m,p), N(m,p)\}$, $A(m,p) = (p/(m+p))(1+\overline{V}(m,p-1))+(m/(m+p))(-1+\overline{V}(m-1,p))$, and $N(m,p) = (p/(m+p))\overline{V}(m,p-1)+(m/(m+p))\overline{V}(m-1,p)$. Hence $A(m,p) = (p-m)/(m+p)+N(m,p)$. Therefore, $\overline{V}(m,p) = A(m,p)$ if $p \geq m$ and $\overline{V}(m,p) = N(m,p)$ if $p < m$. The optimal acceptance policy can now be easily stated as follows: Accept the current drawn ball if the number of $+1$ balls is greater than or equal to the number of -1 balls, otherwise, do not accept the current drawn ball.

Based on the optimal acceptance policy, we will accept the drawn balls until the number of $+1$ balls is less than the number of -1 balls. Since the probability that starting from the position $(m,p)(m \neq p)$ and reaching the position $(i,i)(i > 0$ and $i \leq \min\{m,p\})$ the first time is exactly equal to the probability of starting from the position (p,m) and reaching the position (i,i) the first time, it is easy to see that the following two theorems hold.

THEOREM 2.1. *For any nonnegative integer m and p, $|\overline{V}(m,p) - \overline{V}(p,m)| = |m-p|$.*

THEOREM 2.2. *If $m > p$,*

$$\begin{aligned}\overline{V}(m,p) &= \sum_{j=1}^{p} \overline{V}(j,j) \left\{ \binom{m+p-2j-1}{m-j-1} \right. \\ &\qquad \left. - \binom{m+p-2j-1}{m-j} \right\} \frac{p\cdots(j+1)m\cdots(j+1)}{(p+m)(p+m-1)\cdots(2j+1)} \\ &= \sum_{j=1}^{p} D(j,j) \frac{(m-p)}{(m+p-2j)} \frac{\binom{m+p-2j}{m-j}}{\binom{m+p}{p}}.\end{aligned}$$

Here, $D(i,j) = \binom{i+j}{j}\overline{V}(i,j)$.

THEOREM 2.3. *For any positive integer $m \geq p$,*

$$\overline{V}(m,p) = \sum_{i=1}^{p} \frac{\binom{m+p-2i}{p-i}\binom{2i}{i}}{2\binom{m+p}{p}} = \sum_{i=0}^{p-1} \frac{\binom{m+p}{i}}{\binom{m+p}{p}}$$

$$= p2^{m+p} \int_0^{\frac{1}{2}} x^m (1-x)^{p-1} dx,$$

$$\text{and } \overline{V}(m,m) = \left(2^{2m-1}/\binom{2m}{m}\right) - \frac{1}{2}.$$

Proof. Let X_i be the value of the ith ball $(i = 1,2,\ldots,m+p)$, and let $S_k = \sum_{i=k+1}^{m+p} X_i$ be the kth (tail) partial sum $(k = 0,1,2,...,m+p-1)$. Let $N = \#\{k : S_k = 0,\ 0 \leq k < m+p\}$. Notice that $P(S_{k+1} = 1 \mid S_k = 0) = 1/2$ and that whenever $S_j = 1$, the player gains one unit (according to the optimal policy) by time τ, where $\tau = \min\{k \mid k > j \text{ and } S_k = 0\}$. Hence, $\overline{V}(m,p) = 1/2E(N)$. Notice that each realization of this urn problem is an arrangement of m identical -1 balls and p identical $+1$ balls and that each realization occurs with probability $1/\binom{m+p}{p}$. Thus, $\binom{m+p}{p}E(N) = \sum_w N(w)$, where the sum is taken over all realizations w. Next let T_i

be the number of realizations in which $S_{m+p-2i} = 0$. Since $\sum_w N(w) = \sum_{i=1}^{p} T_i$ and $T_i = \binom{m+p-2i}{p-i}\binom{2i}{i}$, we have $\binom{m+p}{p}E(N) = \sum_{i=1}^{p}\binom{m+p-2i}{p-i}\binom{2i}{i}$. Therefore,

$$\overline{V}(m,p) = \frac{1}{2}E(N) = \sum_{i=1}^{p}\frac{\binom{m+p-2i}{p-i}\binom{2i}{i}}{2\binom{m+p}{p}}.$$

By the combinatorial identity, $\sum_{i=1}^{p}\binom{m+p-2i}{p-i}\binom{2i}{i} = 2\sum_{i=0}^{p-1}\binom{m+p}{i}$; then

$$\overline{V}(m,p) = \sum_{i=0}^{p-1}\frac{\binom{m+p}{i}}{\binom{m+p}{p}}.$$

Since

$$\sum_{i=0}^{l-1}\binom{n}{i}\left(\frac{1}{2}\right)^n = l\binom{n}{i}\int_0^{\frac{1}{2}} x^{n-l}(1-x)^{l-1}dx,$$

then

$$\sum_{i=0}^{p-1}\frac{\binom{m+p}{i}}{\binom{m+p}{p}} = 2^{m+p}p\int_0^{\frac{1}{2}} x^m(1-x)^{p-1}dx.$$

By the combinatorial identity, $\sum_{i=1}^{m}\binom{2m-2i}{m-i}\binom{2i}{i} = 4^m - \binom{2m}{m}$ [5, p. 32]; then

$$\overline{V}(m,m) = \frac{2^{2m-1}}{\binom{2m}{m}} - \frac{1}{2}.$$

The proof of Theorem 2.3 is now complete. □

THEOREM 2.4. *For any positive integer m and p, $D(m,p) = \overline{V}(m,p)\binom{m+p}{p}$ is a positive integer.*

Proof. By Theorem 2.1, it is sufficient to consider the case when $m \geq p$. By Theorem 2.3, $D(m,p) = \overline{V}(m,p)\binom{m+p}{p} = \sum_{i=0}^{p-1}\binom{m+p}{i}$ is a positive integer. □

THEOREM 2.5. *For any nonnegative integer m and p, $\overline{V}(m+1,p+1) > \overline{V}(m,p)$.*

Proof. Since $\overline{V}(m+1,1) > \overline{V}(m,0) = 0$ for any nonnegative integer m, by Theorem 2.1 we can and do assume $m \geq p \geq 1$. By Theorem 2.3,

$$\begin{aligned}
&\overline{V}(m+1,p+1) - \overline{V}(m,p)\\
&= 2^{m+p}\int_0^{\frac{1}{2}} x^m(1-x)^{p-1}\big(4(p+1)x(1-x) - p\big)dx\\
&= 2^{m+p}\int_0^{\frac{1}{2}} x^{m-p+1}\big(4(p+1)x^p(1-x)^p - px^{p-1}(1-x)^{p-1}\big)dx.
\end{aligned}$$

It is sufficient to show that

$$\int_0^{\frac{1}{2}} x^{m-p+1}\big(4(p+1)x^p(1-x)^p - px^{p-1}(1-x)^{p-1}\big)dx > 0.$$

Notice that

$$\begin{aligned}
&\int_0^{\frac{1}{2}}\big(4(p+1)x^p(1-x)^p - px^{p-1}(1-x)^{p-1}\big)dx\\
&= \frac{1}{2}\int_0^1\big(4(p+1)x^p(1-x)^p - px^{p-1}(1-x)^{p-1}\big)dx = \frac{2p!(p+1)!}{(2p+1)!} - \frac{p!p!}{(2p)!} > 0.
\end{aligned}$$

Let x^* be the number in $(0,1/2)$ such that $4(p+1)x^*(1-x^*)=p$. Then $4(p+1)x(1-x)-p\le 0$ if $0\le x\le x^*$ and $4(p+1)x(1-x)-p\ge 0$ if $x^*\le x\le 1/2$. Hence,

$$\begin{aligned}&\int_0^{\frac{1}{2}}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx\\&\quad=\int_{x^*}^{\frac{1}{2}}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx\\&\quad-\int_0^{x^*}\big(px^{p-1}(1-x)^{p-1}-4(p+1)x^p(1-x)^p\big)dx>0;\end{aligned}$$

that is,

$$\begin{aligned}&\int_{x^*}^{\frac{1}{2}}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx\\&\quad>\int_0^{x^*}\big(px^{p-1}(1-x)^{p-1}-4(p+1)x^p(1-x)^p\big)dx.\end{aligned}$$

By the Mean Value theorem,

$$\begin{aligned}&\int_0^{\frac{1}{2}}x^{m-p+1}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx\\&\quad=x_2^{m-p+1}\int_{x^*}^{\frac{1}{2}}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx\\&\quad-x_1^{m-p+1}\int_0^{x^*}\big(px^{p-1}(1-x)^{p-1}-4(p+1)x^p(1-x)^p\big)dx,\end{aligned}$$

where $0\le x_1\le x^*\le x_2\le 1/2$. Hence,

$$\int_0^{\frac{1}{2}}x^{m-p+1}\big(4(p+1)x^p(1-x)^p-px^{p-1}(1-x)^{p-1}\big)dx>0,$$

since $m\ge p$. Therefore, $\overline{V}(m+1,p+1)>\overline{V}(m,p)$ for all nonnegative integers m and p. □

THEOREM 2.6.

1. $1/(m+p+1)\le\overline{V}(m,p+1)-\overline{V}(m,p)\le 1$.
2. $0\le\overline{V}(m,p)-\overline{V}(m+1,p)\le 1-(1/(m+p+1))$.

Proof. By Theorems 2.1 and 2.3, it is easy to check that $\overline{V}(m,p)-\overline{V}(m+1,p)\ge 0$ and to also check that $1/(m+p+1)\le\overline{V}(m,p+1)-\overline{V}(m,p)$ is equivalent to that $\overline{V}(m,p)-\overline{V}(m+1,p)\le 1-(1/(m+p+1))$ by Theorem 2.1. Theorem 2.6 is clearly true when $n=m+p=1$. Now, by mathematical induction on n, we can prove Theorem 2.6 easily (details are omitted). □

THEOREM 2.7. *For any positive integer k, $\overline{V}(km,m)$ and $\overline{V}(m,km)$ are strictly increasing in m.*

Proof. By Theorem 2.1, it is sufficient to show that $\overline{V}(km,m)$ is strictly increasing in m. By Theorem 2.5, Theorem 2.7 holds when $k=1$. Now we will prove Theorem 2.7 when $k\ge 2$. By Theorem 2.3,

$$\begin{aligned}&\overline{V}(k(m+1),m+1)-\overline{V}(km,m)\\&\quad=2^{km+m}\int_0^{\frac{1}{2}}x^{km}(1-x)^{m-1}\big(2^{k+1}(m+1)x^k(1-x)-m\big)dx\\&\quad=2^{km+m}\int_0^{\frac{1}{2}}x^{m-1}(1-x)^{m-1}x^{km-m+1}\big(2^{k+1}(m+1)x^k(1-x)-m\big)dx.\end{aligned}$$

Since $m \geq 1$, $x^{m-1}(1-x)^{m-1}$ is strictly increasing and nonnegative on the interval $[0, 1/2]$, $2^{k+1}(m+1)x^k(1-x) - m \leq 0$ if $0 \leq x \leq x^*$, ≥ 0 if $x* \leq x \leq 1/2$, where $0 < x^* < 1/2$. By the Mean Value theorem, it is sufficient to show that

$$\int_0^{\frac{1}{2}} x^{km-m+1}\big(2^{k+1}(m+1)x^k(1-x) - m\big)dx > 0.$$

By a direct computation,

$$\int_0^{\frac{1}{2}} x^{km-m+1}\big(2^{k+1}(m+1)x^k(1-x) - m\big)dx$$
$$= (1/2)^{km-m+2}\left(\frac{2(m+1)}{km+k-m+2} - \frac{m+1}{km+k-m+3} - \frac{m}{km-m+2}\right) > 0,$$

since $k \geq 2$. Therefore, $\overline{V}(k(m+1), m+1) - \overline{V}(km, m) > 0$, and the proof of Theorem 2.7 now is complete. □

3. Asymptotic behavior of $\overline{V}(m, p)$. By Theorems 2.1, 2.2, and 2.3, we have an exact closed form solution for $\overline{V}(m, p)$. However, it is only useful when m or p is small. In this section, we will derive some asymptotic forms for $\overline{V}(m, p)$ when m and $p \to \infty$.

THEOREM 3.1. *$\overline{V}(m, p) \to p/(m-p)$ if $m/p \to \lambda > 1$.*

Proof. By Theorem 2.3,

$$\begin{aligned}\overline{V}(m,p) &= \sum_{i=1}^{p} \frac{\binom{m+p-2i}{p-i}\binom{2i}{i}}{2\binom{m+p}{p}} \\ &\sim \frac{1}{2}\sum_{\gamma=1}^{\infty}\binom{2\gamma}{\gamma}\left(\frac{\lambda}{(1+\lambda)^2}\right)^{\gamma} \\ &= \frac{1}{\lambda-1} = \frac{p}{m-p}\end{aligned}$$

if $m/p \to \lambda > 1$. □

THEOREM 3.2.

1. *$\overline{V}(m,p)/\sqrt{p/2} \to \exp(\alpha^2/2)\int_\alpha^\infty \exp(-t^2/2)dt$ if $(m-p)/\sqrt{2p} \to \alpha \geq 0$ as $m, p \to \infty$;*
2. *$\overline{V}(m,p)/\sqrt{p/2} \to 2\alpha + \exp(\alpha^2/2)\int_\alpha^\infty \exp(-t^2/2)dt$ if $(m-p)/\sqrt{2p} \to -\alpha \leq 0$ as $m, p \to \infty$;*
3. *For any integer k, $\overline{V}(k+p, p)/\big(\sqrt{\pi p}/2\big) \to 1$ as $p \to \infty$.*

Proof. By Theorem 2.3, for $m \geq p$,

$$\overline{V}(m,p) = \sum_{k=0}^{p-1}\frac{\binom{m+p}{i}}{\binom{m+p}{p}} = P(X \leq p-1)/P(X = p),$$

where X is a binomial random variable with parameters $m+p$ and $1/2$. By the central limit theorem [1, p. 42],

$$\frac{P(X \leq p-1)/P(X=p)}{\sqrt{p/2}} \to \exp(\alpha^2/2)\int_\alpha^\infty \exp(-t^2/2)dt$$

if $(m-p)/\sqrt{2p} \to \alpha \geq 0$ as $m, p \to \infty$.

By Theorem 2.1, for $m < p$, $\overline{V}(m,p) = \overline{V}(p,m) + p - m$. Then, by the same argument,

$$\frac{\overline{V}(m,p)}{\sqrt{p/2}} = \frac{(p-m)}{\sqrt{p/2}} + \frac{\overline{V}(p,m)}{\sqrt{p/2}} \to 2\alpha + \exp(\alpha^2/2)\int_\alpha^\infty \exp(-t^2/2)dt$$

if $(m-p)/\sqrt{2p} \to -\alpha \le 0$ as $m, p \to \infty$.

When $\alpha = 0$, $\int_\alpha^\infty \exp(-t^2/2)dt = \sqrt{\pi/2}$. Hence, $\overline{V}(k+p,p)/\big(\sqrt{\pi p}/2\big) \to 1$ as $p \to \infty$. The proof of Theorem 3.2 is now complete. □

4. The original (m,p) urn problem. For any nonnegative integer m and p, let $V(m,p)$ be the value of the original (m,p) urn problem proposed by Shepp as stated in section 1. We now want to compare $V(m,p)$ and $\overline{V}(m,p)$.

THEOREM 4.1. *$V(m,0) = \overline{V}(m,0)$ for all $m = 0,1,2,\ldots$ and $V(0,p) = \overline{V}(0,p) = p$ and $V(1,p) = \overline{V}(1,p) = p^2/(1+p)$ for all $p = 0,1,2,\ldots$.*

Proof. Since, when $p = 0$ or $m = 0$ or 1 two problems are the same, they have the same value. □

THEOREM 4.2. *For any positive integer $m \ge 2$ and $p \ge 1, V(m,p) < \overline{V}(m,p)$.*

Proof. For any positive integer m and p,

$$V(m,p) = \max\left\{0, \frac{p-m}{p+m} + \frac{m}{p+m}V(m-1,p) + \frac{p}{p+m}V(m,p-1)\right\}$$

and

$$\overline{V}(m,p) = \frac{(p-m)^+}{p+m} + \frac{m}{p+m}\overline{V}(m-1,p) + \frac{p}{p+m}\overline{V}(m,p-1).$$

By Theorem 4.1, $V(m,0) = \overline{V}(m,0)$for all $m = 0,1,2,\ldots$ and $V(1,p) = \overline{V}(1,p) = p^2/(p+1)$ for all $p = 0,1,2,\ldots$.

Now by mathematical induction we can conclude that $\overline{V}(m,p) > V(m,p)$ for any integers $m \ge 2$ and $p \ge 1$ since $\overline{V}(2,1) = 1/3 > V(2,1) = 0$. □

For the original (m,p) urn problem, if

$$E(m+1,p) = \frac{m+1}{m+1+p}\big(-1 + V(m,p)\big) + \frac{p}{m+1+p}\big(1 + V(m+1,p-1)\big) \ge 0,$$

then $V(m,p) - V(m+1,p) \ge 1/(m+1+p)$. However, for the acceptance (m,p) urn problem, we do not have this inequality. For instance, $\overline{V}(1,1) - \overline{V}(2,1) = 1/2 - 1/3 = 1/6 < 1/3$.

In the original (m,p) urn problem, the last ball drawn, under the optimal drawing policy, is always a $+1$ ball. Similarly, we have the following theorem in the acceptance (m,p) urn problem.

THEOREM 4.3. *In the acceptance (m,p) urn problem, the last ball accepted under the optimal acceptance policy is always a $+1$ ball.*

Proof. Under the optimal acceptance policy, one will accept the current drawn ball if and only if the number of $+1$ balls is greater than or equal to the number of -1 balls. Now if the current drawn one is a -1 ball, then the number of $+1$ balls will be still greater than the number of -1 balls. Hence, the player will accept the

next drawn ball until he gets a +1 ball. Thus, a −1 ball is never the last accepted ball. □

THEOREM 4.4.

$$\lim_{p\to\infty}(V(m,p+1)-V(m,p))=\lim_{p\to\infty}(V(m,p)-V(m+1,p))=1,$$

$$\lim_{p\to\infty}(\overline{V}(m,p+1)-\overline{V}(m,p))=\lim_{p\to\infty}(\overline{V}(m,p)-\overline{V}(m+1,p))=1.$$

Proof. Since $\lim_{m\to\infty}\overline{V}(m,p)=0$ for any fixed p,

$$\lim_{p\to\infty}\big(\overline{V}(m,p+1)-\overline{V}(m,p)\big)=1+\lim_{p\to\infty}\big(\overline{V}(p+1,m)-\overline{V}(p,m)\big)=1.$$

Similarly,

$$\lim_{p\to\infty}\big(\overline{V}(m,p)-\overline{V}(m+1,p)\big)=1+\lim_{p\to\infty}\big(\overline{V}(p,m)-\overline{V}(p,m+1)\big)=1. \qquad \square$$

For any nonnegative integer m and p, define

$$\begin{aligned}\Delta^2V_p(m)&=V(m+2,p)+V(m,p)-2V(m+1,p),\\ \Delta^2V_m(p)&=V(m,p+2)+V(m,p)-2V(m,p+1),\\ \Delta^2V(m,p)&=V(m+2,p)+V(m,p+2)-2V(m+1,p+1),\end{aligned}$$

and define $\Delta^2\overline{V}_p(m)$, $\Delta^2\overline{V}_m(p)$, and $\Delta^2\overline{V}(m,p)$ accordingly.

In [4], Chen and Hwang proved that $\Delta^2V_p(m)\ge 0$, $\Delta^2V_m(p)\ge 0$, and $\Delta^2V(m,p)\ge 0$. The next theorem shows that $\Delta^2\overline{V}_p(m)>0$, $\Delta^2\overline{V}_m(p)>0$, and $\Delta^2\overline{V}(m,p)>0$, for all positive integers m and p.

THEOREM 4.5. *For any positive integer m and p, $\Delta^2\overline{V}_m(p)>0$, $\Delta^2\overline{V}_p(m)>0$, and $\Delta^2\overline{V}(m,p)>0$.*

Proof. By definition, $\Delta^2\overline{V}_p(m)=\overline{V}(m+2,p)+\overline{V}(m,p)-2\overline{V}(m+1,p)$.

Case 1. Suppose that $m\ge p$; then by Theorem 2.3,

$$\begin{aligned}\Delta^2\overline{V}_p(m)&=2^{m+p+2}p\int_0^{\frac12}x^{m+2}(1-x)^{p-1}dx+2^{m+p}p\int_0^{\frac12}x^m(1-x)^{p-1}dx\\ &\quad-2^{m+p+2}p\int_0^{\frac12}x^{m+1}(1-x)^{p-1}dx\\ &=2^{m+p}p\int_0^{\frac12}x^m(1-x)^{p-1}(4x^2-4x+1)dx>0.\end{aligned}$$

since $m\ge 1$ and $p\ge 1$.

Case 2. Suppose that $p=m+1$; then by Theorems 2.1 and 2.3,

$$\begin{aligned}\Delta^2\overline{V}_p(m)&=\overline{V}(m+2,m+1)+\overline{V}(m,m+1)-2\overline{V}(m+1,m+1)\\ &=\overline{V}(m+2,m+1)+\overline{V}(m+1,m)+1-2\overline{V}(m+1,m+1)\\ &=\frac{2^{2m+2}}{\binom{2m+3}{m+1}}+\frac{2^{2m}}{\binom{2m+1}{m}}-\frac{2^{2m+2}}{\binom{2m+2}{m+1}}\\ &=2^{2m+1}(m+1)!(m+1)!/(2m+3)!>0.\end{aligned}$$

Case 3. Suppose that $p \geq m+2$; then by Theorems 2.1 and 2.3,

$$\begin{aligned}\Delta^2\overline{V}_p(m) &= \overline{V}(m+2,p) + \overline{V}(m,p) - 2\overline{V}(m+1,p)\\ &= \overline{V}(p,m+2) + \overline{V}(p,m) - 2\overline{V}(p,m+1)\\ &= (m+2)2^{m+p+2}\int_0^{\frac{1}{2}} x^p(1-x)^{m+1}dx + m2^{m+p}\int_0^{\frac{1}{2}} x^p(1-x)^{m-1}dx\\ &\quad -(m+1)2^{m+p+2}\int_0^{\frac{1}{2}} x^p(1-x)^m dx\\ &= 2^{m+p}\int_0^{\frac{1}{2}} x^p(1-x)^{m-1}\big(4(m+2)(1-x)^2 - 4(m+1)(1-x) + m\big)dx.\end{aligned}$$

Notice that $g(m) = 4(m+2)(1-x)^2 - 4(m+1)(1-x) + m$ is strictly increasing in m for all $0 \leq x \leq 1/2$ and $g(0) \geq 0$ if $0 \leq x \leq 1/2$. Hence,

$$2^{m+p}\int_0^{\frac{1}{2}} x^p(1-x)^{m-1}\big(4(m+2)(1-x)^2 - 4(m+1)(1-x) + m\big)dx > 0.$$

$\Delta^2\overline{V}_m(p) > 0$ and $\Delta^2\overline{V}(m,p) > 0$ can be proved similarly. □

Based on Theorems 2.1, 2.3, and 2.5, we can also prove the following interesting theorems.

THEOREM 4.6. *For any nonnegative integer m and p,*

$$\begin{aligned}2\overline{V}(m,p) &< \overline{V}(m,p) + \overline{V}(m+1,p+1)\\ &< \overline{V}(m+1,p) + \overline{V}(m,p+1) \leq 2\overline{V}(m+1,p+1).\end{aligned}$$

Proof. By Theorem 2.7, $\overline{V}(m,p) < \overline{V}(m+1,p+1)$, and $2\overline{V}(m,p) < \overline{V}(m,p) + \overline{V}(m+1,p+1)$.

Case 1. If $m = 0$, then

$$\overline{V}(m+1,p) + \overline{V}(m,p+1) = \overline{V}(1,p) + \overline{V}(0,p+1) = p + 1 + p^2/(p+1),$$

$$\overline{V}(m,p) + \overline{V}(m+1,p+1) = \overline{V}(0,p) + \overline{V}(1,p+1) = p + (p+1)^2/(p+2),$$

and

$$2\overline{V}(m+1,p+1) = 2\overline{V}(1,p+1) = 2(p+1)^2/(p+2).$$

It is easy to see that

$$\overline{V}(m,p) + \overline{V}(m+1,p+1) < \overline{V}(m+1,p) + \overline{V}(m,p+1) < 2\overline{V}(m+1,p+1).$$

Case 2. If $p = 0$, then

$$\overline{V}(m,p) + \overline{V}(m+1,p+1) = \overline{V}(m,0) + \overline{V}(m+1,1) = 1/(m+2),$$

$$\overline{V}(m+1,p) + \overline{V}(m,p+1) = \overline{V}(m+1,0) + \overline{V}(m,1) = 1/(m+1),$$

and

$$2\overline{V}(m+1,p+1) = 2\overline{V}(m+1,1) = 2/(m+2).$$

Hence,

$$\overline{V}(m,p)+\overline{V}(m+1,p+1)<\overline{V}(m+1,p)+\overline{V}(m,p+1)\le\overline{V}(m+1,p+1).$$

Now we assume that $m\ge 1$ and $p\ge 1$.

Case 3. If $m\ge p+1$, then by Theorem 2.3

$$\overline{V}(m,p)-\overline{V}(m+1,p)=2^{m+p}p\int_0^{\frac{1}{2}}x^m(1-x)^{p-1}(1-2x)dx$$

and

$$\begin{aligned}&\overline{V}(m,p+1)-\overline{V}(m+1,p+1)\\&\quad=2^{m+p+1}(p+1)\int_0^{\frac{1}{2}}x^m(1-x)^p(1-2x)dx\\&\quad=2^{m+p}(p+1)\int_0^{\frac{1}{2}}x^m(1-x)^{p-1}(1-2x)(2-2x)dx\\&\quad>2^{m+p}p\int_0^{\frac{1}{2}}x^m(1-x)^{p-1}(1-2x)dx.\end{aligned}$$

Hence,

$$\overline{V}(m,p)+\overline{V}(m+1,p+1)<\overline{V}(m+1,p)+\overline{V}(m,p+1).$$

On the other hand,

$$\begin{aligned}&2\overline{V}(m+1,p+1)-\overline{V}(m+1,p)-\overline{V}(m,p+1)\\&=((m-p)/(m+p+2))\overline{V}(m,p+1)-((m-p)/(m+p+2))\overline{V}(m+1,p)>0.\end{aligned}$$

Hence,

$$\overline{V}(m,p)+\overline{V}(m+1,p+1)<\overline{V}(m+1,p)+\overline{V}(m,p+1)<2\overline{V}(m+1,p+1).$$

Case 4. If $m=p$, then

$$\overline{V}(m+1,p+1)=\overline{V}(m+1,m+1)=1/2\ \overline{V}(m+1,m)+1/2\ \overline{V}(m,m+1).$$

Hence,

$$\overline{V}(m,m)+\overline{V}(m+1,m+1)<2\overline{V}(m+1,m+1)=\overline{V}(m,m+1)+\overline{V}(m,m+1).$$

Case 5. If $m<p$, then

$$\overline{V}(m+1,p)+\overline{V}(m,p+1)=\overline{V}(p,m+1)+\overline{V}(p+1,m)+2p-2m,$$

$$\overline{V}(m,p)+\overline{V}(m+1,p+1)=\overline{V}(p,m)+\overline{V}(p+1,m+1)+2p-2m,$$

and

$$2\overline{V}(m+1,p+1)=2\overline{V}(p+1,m+1)+2p-2m.$$

By Case 3,

$$\overline{V}(m,p)+\overline{V}(m+1,p+1)<\overline{V}(m+1,p)+\overline{V}(m,p+1)<2\overline{V}(m+1,p+1).$$

The proof of Theorem 4.6 is now complete. □

By Theorem 2.5,

$$\overline{V}(m,p) < \overline{V}(m+1,p+1) < \overline{V}(m+2,p+2)$$

for all nonnegative integers m and p. The next theorem reveals that for all nonnegative integers m and p, $\overline{V}(m+k,p+k)$ is a concave function of k.

THEOREM 4.7. *For any nonnegative integer m and p, $\overline{V}(m,p)+\overline{V}(m+2,p+2) < 2\overline{V}(m+1,p+1)$.*

Proof. By Theorem 2.1,

$$\overline{V}(m,p)+\overline{V}(m+2,p+2)-2\overline{V}(m+1,p+1) = \overline{V}(p,m)+\overline{V}(p+2,m+2)-2\overline{V}(p+1,m+1)$$

if $m < p$. Since it is easy to see that Theorem 4.7 is true when $p = 0$, we will assume $m \geq p \geq 1$ in the following proof. Now for any positive integer m and p, we write $\overline{V}(m,p) = \overline{V}(n+p,p)$, where $n = m - p \geq 0$. By Theorem 2.3,

$$\begin{aligned}\overline{V}(n+p,p) &= 2^{n+2p}p\int_0^{\frac{1}{2}} x^{n+p}(1-x)^{p-1}dx \\ &= 2^n\int_0^{\frac{1}{2}} x^n(1-x)^{-1}p\big(4x(1-x)\big)^p dx \\ &= \frac{1}{2}\int_0^1 g(t)pt^p dt,\end{aligned}$$

where $g(t) = \big(1-\sqrt{1-t}\big)^n\big(1+\sqrt{1-t}\big)^{-1}(1-t)^{-1/2}$. Hence,

$$\overline{V}(n+p+2,p+2) - 2\overline{V}(n+p+1,p+1) + \overline{V}(n+p,p) = \frac{1}{2}\int_0^1 g(t)h(t)dt,$$

where $h(t) = (p+2)t^{p+2} - 2(p+1)t^{p+1} + pt^p$. Notice that $h(t) \geq 0$ if $0 \leq t \leq p/(p+2)$ and $h(t) \leq 0$ if $p/(p+2) \leq t \leq 1$. Also notice that

$$\int_0^1 h(t)dt = (p+2)/(p+3) - 2(p+1)/(p+2) + p/(p+1) < 0.$$

Hence,

$$\begin{aligned}&\overline{V}(n+p+2,p+2) + \overline{V}(n+p,p) - 2\overline{V}(n+p+1,p+1) \\ &\qquad = \frac{1}{2}\int_0^1 g(t)h(t)dt = \frac{1}{2}\int_0^{t^*} g(t)h(t)dt + \frac{1}{2}\int_{t^*}^1 g(t)h(t)dt,\end{aligned}$$

where $t^* = p/(p+2)$. Hence, by the Mean Value theorem, $1/2\int_0^{t^*} g(t)h(t)dt = 1/2g(t_1)\int_0^{t^*} h(t)dt$ and $1/2\int_{t^*}^1 g(t)h(t)dt = 1/2g(t_2)\int_{t^*}^1 h(t)dt$, where $0 < t_1 < t^* < t_2 < 1$. Since g is strictly increasing in t, $0 \leq t \leq 1$, $0 < g(t_1) < g(t_2)$. Since, $0 < \int_0^{t^*} h(t)dt < -\int_{t^*}^1 h(t)dt$, $g(t_1)\int_0^{t^*} h(t)dt < -g(t_2)\int_{h^*}^1 h(t)dt$. Therefore,

$$\begin{aligned}&\overline{V}(n+p+2,p+2) + \overline{V}(n+p,p) - 2\overline{V}(n+p+1,p+1) \\ &\qquad = \frac{1}{2}g(t_1)\int_0^{t^*} h(t)dt + \frac{1}{2}g(t_2)\int_{t^*}^1 h(t)dt < 0,\end{aligned}$$

and the proof of Theorem 4.7 is now complete. □

5. A variation of the acceptance (m, p) urn problem. In the stock market, investors try to sell if the future price will go down and try to buy if the future price will go up, so the following variation of the acceptance (m, p) urn problem will be a suitable model.

An urn contains m balls of value -1 and p balls of value $+1$. Each turn a ball is drawn randomly, without replacement, and the player decides before the draw whether or not to accept and guess the ball. If he accepts and guesses correctly he gets a $+1$; if he accepts and guesses incorrectly he gets a -1. The process continues until all $m + p$ balls are drawn.

Let $W(m, p)$ denote the value of this new variation. Let $A_0(m, p)$ be the expected value of accepting the current drawn ball from the (m, p) urn and guessing it is a -1 ball, assuming an optimal accepting and guessing policy is followed after the current one. Let $A_1(m, p)$ be the expected value of accepting the current drawn ball from the (m, p) urn and guessing it is a $+1$ ball, assuming an optimal accepting and guessing policy is followed after this one. Let $A(m, p) = \max\{A_0(m, p), A_1(m, p)\}$, and let $N(m, p)$ be the expected value of not accepting the current drawn ball from the (m, p) urn, assuming an optimal accepting and guessing policy is followed. It is obvious that $W(m, p) = \max\{A(m, p), N(m, p)\}$. Since $A_0(m, p) = (m/(m + p))\big(1 + W(m - 1, p)\big) + (p/(m + p))\big(-1 + W(m, p - 1)\big)$ and $A_1(m, p) = (m/(m + p))\big(-1 + W(m - 1, p)\big) + (p/(m + p))\big(1 + W(m, p - 1)\big)$, $A_0(m, p) < A_1(m, p), = A_1(m, p)$, or $> A_1(m, p)$ accordingly as $m > p = p$, or $< p$. Hence,

$$\begin{aligned} A(m, p) &= (1/(m + p))\big(|m - p| + mW(m - 1, p) + pW(m, p - 1)\big) \\ &\geq N(m, p) = (1/(m + p))\big(mW(m - 1, p) + pW(m, p - 1)\big), \end{aligned}$$

since $|m - p| \geq 0$. Therefore $W(m, p) = A(m, p) = (1/(m + p))\big(|m - p| + mW(m - 1, p) + pW(m, p - 1)\big)$. The optimal guessing policy is to guess that it is a -1 ball if $m > p$, guess that it is a $+1$ ball if $m < p$, and guess randomly if $m = p$. If balls of value $+1$ mean that the price will go up and balls of value -1 mean that the price will go down, then guessing $+1$ means to buy and guessing -1 means to sell. The optimal guessing policy is consistent with the optimal practice of investors. The following theorems can be proved.

THEOREM 5.1. *For any nonnegative integer i and j, $W(i, j) = W(j, i)$.*

THEOREM 5.2. *For any nonnegative integer i and j, $W(i, j) = \overline{V}(i, j) + \overline{V}(j, i)$.*

6. A Bayesian approach to the acceptance (m, p) urn problem. In a financial or marketing problem, the total number of balls is usually known but the number of balls of value -1 is unknown and is a random variable. A Bayesian approach to this optimal stopping problem would be appropriate.

Now let $n = m + p$ be the total number of balls in the urn, and let θ be the initial prior distribution of the random variable m (number of balls of value -1). Let $N_n(\theta)$ denote the expected value of not accepting the current drawn ball from the urn, assuming an optimal Bayesian acceptance policy is followed, and let $A_n(\theta)$ denote the expected value of accepting the current drawn ball from the urn, assuming an optimal Bayesian acceptance policy is followed. Let $\overline{V}_n(\theta) = \max\{N_n(\theta), A_n(\theta)\}$ denote the value of the urn with n balls and the prior distribution θ.

Let x_1 be the value of the first drawn ball. It is easy to see that $A_n(\theta) = \int\big(x_1 + \overline{V}_{n-1}(\theta(x_1))\big)\theta(dx_1)$ and $N_n(\theta) = \int \overline{V}_{n-1}(\theta(x_1))\theta(dx_1)$. Here $\theta(x_1)$ is the posterior distribution of the number of balls of value -1 after the first draw given that $X_1 = x_1$. Since $A_n(\theta) \geq N_n(\theta)$ if and only if $\int x_1\theta(dx_1) = \theta(X_1 = 1) - \theta(X_1 = -1) \geq 0$, one would accept the current drawn ball if $\theta(X_1 = 1) \geq \theta(X_1 = -1)$. Therefore, the

optimal Bayesian acceptance policy can be simply stated as follows: for $k = 1, 2, \ldots, n$, the player will accept the kth drawn ball if and only if $\theta(X_k = 1 \mid x_1, x_2, \ldots, x_{k-1}) \geq \theta(X_k = -1 \mid x_1, x_2, \ldots, x_{k-1})$ where $\theta(\cdot \mid x_1, \ldots, x_{k-1})$ is the posterior distribution of the number of -1 balls given that $X_1 = x_1, X_2 = x_2, \ldots, X_{k-1} = x_{k-1}$.

Now suppose that the initial prior distribution θ of m (the number of -1 balls) is uniform over the set $\{0, 1, 2, \ldots, n\}$. Since $\sum_{i=1}^{k} X_i$ is a sufficient statistic for the unknown parameter m, $\theta(X_k = 1 \mid \sum_{i=1}^{k-1} X_i) \geq \theta(X_k = -1 \mid \sum_{i=1}^{k-1} X_i)$ if an only if $\sum_{i=1}^{k-1} X_i \geq 0$. The player will accept the kth drawn ball if and only if $\sum_{i=1}^{k-1} X_i \geq 0$. It is worth noticing that the character of the optimal Bayesian acceptance policy is similar to that of the optimal acceptance policy of the non-Bayesian urn problem. However, when m is known, under the optimal acceptance policy the ball accepted last is always a $+1$, but under an optimal Bayesian acceptance policy the ball accepted last is always a -1 except for the nth ball.

The following are values of $\overline{V}_n(\theta)$ when θ is uniform:

$$\begin{aligned} &n = 1, && \overline{V}_n(\theta) = 0, \\ &n = 2, && \overline{V}_n(\theta) = 1/6, \\ &n = 3, && \overline{V}_n(\theta) = 1/3, \\ &n = 4, && \overline{V}_n(\theta) = 17/30. \end{aligned}$$

Notice that $E(m \mid n = 2) = 1$, but $\overline{V}_2(\theta) = 1/6 < \overline{V}(1, 1) = 1/2$; $E(m \mid n = 4) = 2$, but $\overline{V}_4(\theta) = 17/30 < \overline{V}(2, 2) = 5/6$. These facts are expected since we have full information about an acceptance (m, p) urn and we have only partial information about a random acceptance (m, p) urn, i.e., when m is a random variable. Furthermore, $\overline{V}_n(\theta)$ is nondecreasing in n since the player has more times to decide whether or not to accept.

7. Application and numerical illustration. The acceptance (m, p) urn model studied above can be useful in the following financial situation. Suppose that we expect there will be m downs and p ups in the stock price (or bond price). Suppose that the up or down will be on an equal scale. We buy the stock and sell it at the next time unit. If the price goes up one unit we make a profit; otherwise we lose. Our goal is to maximize the gain. Based on our acceptance (m, p) urn model, we should buy the stock if and only if the number of the ups is greater than the number of the downs. Otherwise we should not have any trading.

The variation of the acceptance (m, p) urn model discussed in section 5 can be used in the following situation. Suppose that we expect that there will be m downs and p ups in the stock price. If we know the price will be up, certainly we should buy the stock and sell later. If we know the price will be down, we should sell the stock and buy back later. Our goal is to maximize the gain between "in and out." The optimal strategy will be that "buy now sell later" if the number of the ups is greater than the number of the downs; conversely, "sell now and buy back later" if the number of the ups is less than the number of the downs.

Certainly, the numbers of the ups and downs are not known, and they are random. Therefore, the Bayesian approach to the acceptance (m, p) urn model would be much more suitable to the financial application. The details will be presented in another article.

The following three tables of values of $V(m, p)$, $\overline{V}(m, p)$, and $W(m, p)$ are given for the sake of comparison.

Acknowledgment. We would like to thank the referee for his invaluable comments which led to a simpler and more intuitive proof of Theorem 2.3, and also for correcting a mistake in Theorem 3.2.

TABLE 1
$V(m,p)$.

p(plus)	m(minus)									
	0	1	2	3	4	5	6	7	8	9
9	9	8.10	7.20	6.31	5.43	4.58	3.75	2.95	2.21	1.53
8	8	7.11	6.22	5.35	4.49	3.66	2.86	2.11	1.43	0.84
7	7	6.13	5.25	4.39	3.56	2.76	2.01	1.34	0.66	0.23
6	6	5.14	4.29	3.45	2.66	1.91	1.23	0.66	0.23	0
5	5	4.17	3.33	2.54	1.79	1.12	0.55	0.15	0	0
4	4	3.20	2.40	1.66	1.00	0.44	0.07	0	0	0
3	3	2.25	1.50	0.85	0.34	0	0	0	0	0
2	2	1.33	0.67	0.20	0	0	0	0	0	0
1	1	0.50	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

TABLE 2
$\overline{V}(m,p)$.

p(plus)	m(minus)									
	0	1	2	3	4	5	6	7	8	9
9	9	8.10	7.22	6.36	5.53	4.73	3.99	3.30	2.70	2.20
8	8	7.11	6.24	5.41	4.60	3.85	3.16	2.55	2.05	1.70
7	7	6.13	5.28	4.47	3.70	3.00	2.39	1.89	1.55	1.30
6	6	5.14	4.32	3.55	2.83	2.22	1.72	1.39	1.16	0.99
5	5	4.17	3.38	2.66	2.03	1.53	1.22	1.00	0.85	0.73
4	4	3.20	2.47	1.83	1.33	1.03	0.83	0.70	0.60	0.53
3	3	2.25	1.60	1.10	0.83	0.66	0.55	0.47	0.41	0.36
2	2	1.33	0.83	0.60	0.47	0.38	0.32	0.28	0.24	0.22
1	1	0.50	0.33	0.25	0.20	0.17	0.14	0.13	0.11	0.10
0	0	0	0	0	0	0	0	0	0	0

TABLE 3
$W(m,p)$.

p(plus)	m(minus)									
	0	1	2	3	4	5	6	7	8	9
9	9	8.20	7.44	6.72	6.06	5.46	4.98	4.60	4.40	4.40
8	8	7.22	6.48	5.82	5.20	4.70	.4.32	4.10	4.10	4.40
7	7	6.26	5.56	4.94	4.40	4.00	3.78	3.78	4.10	4.60
6	6	5.28	4.64	4.10	3.66	3.44	3.44	3.78	4.32	4.98
5	5	4.34	3.76	3.32	3.06	3.06	3.44	4.00	4.70	5.46
4	4	3.40	2.94	2.66	2.66	3.06	3.66	4.40	5.20	6.06
3	3	2.50	2.20	2.20	2.66	3.32	4.10	4.94	5.82	6.72
2	2	1.66	1.66	2.20	2.94	3.76	4.64	5.56	6.48	7.44
1	1	1.00	1.66	2.50	3.40	4.34	5.28	6.26	7.22	8.20
0	0	1	2	3	4	5	6	7	8	9

REFERENCES

[1] P. BILLINGSLEY, *Convergence of Probability Measures*, John Wiley and Sons, New York, 1968.
[2] W. M. BOYCE, *Stopping rules for selling bonds*, Bell J. Econ. Manage Sci., 1 (1970), pp. 27–53.
[3] W. M. BOYCE, *On a simple optimal stopping problem*, Discrete Math., 5 (1973), pp. 297–312.
[4] R. W. CHEN AND F. K. HWANG, *On the values of an (m,p) urn*, Congr. Numer., 41 (1984), pp. 75–84.
[5] H. W. GOULD, *Combinatorial Identities*, Morgantown, WV, 1972.
[6] N. L. JOHNSON AND S. KOTZ, *Urn Models and Their Applications*, John Wiley and Sons, New York, 1977.
[7] L. A. SHEPP, *Explicit solutions to some problems of optimal stopping*, Ann. Math. Statist., 40 (1969), pp. 993–1010.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 196–204, May 1998

002

THE LOVÁSZ THETA FUNCTION AND A SEMIDEFINITE PROGRAMMING RELAXATION OF VERTEX COVER*

JON KLEINBERG† AND MICHEL X. GOEMANS‡

Abstract. Let $vc(G)$ denote the minimum size of a vertex cover of a graph $G = (V, E)$. It is well known that one can approximate $vc(G)$ to within a factor of 2 in polynomial time; and despite considerable investigation, no $(2-\varepsilon)$-approximation algorithm has been found for any $\varepsilon > 0$. Because of the many connections between the independence number $\alpha(G)$ and the Lovász theta function $\vartheta(G)$, and because $vc(G) = |V| - \alpha(G)$, it is natural to ask how well $|V| - \vartheta(G)$ approximates $vc(G)$. It is not difficult to show that these quantities are within a factor of 2 of each other ($|V| - \vartheta(G)$ is never less than the value of the canonical linear programming relaxation of $vc(G)$); our main result is that $vc(G)$ can be more than $(2 - \varepsilon)$ times $|V| - \vartheta(G)$ for any $\varepsilon > 0$. We also investigate a stronger lower bound than $|V| - \vartheta(G)$ for $vc(G)$.

Key words. vertex cover, independent sets, approximation algorithms, semidefinite programming.

AMS subject classifications. 90C27, 90C25, 68Q25, 05B99

PII. S0895480195287541

1. Introduction.

Let $G = (V, E)$ be an undirected graph. By a vertex cover of G we mean a set $S \subset V$ such that for each $e \in E$ at least one endpoint of e lies in S. Thus, a vertex cover is the complement of an independent set in G. For a graph in which each vertex i is given a nonnegative weight w_i, the problem of finding a vertex cover of minimum total weight is a classical NP-complete problem. We are interested here in the question of finding approximate solutions to this problem in polynomial time.

We can formulate the problem of finding a minimum-weight vertex cover via the following integer program. Assign a variable x_i to each vertex $i \in V$; then we have

$$\begin{array}{lll} \text{(VC)} \quad \text{Min} & \sum_i w_i x_i & \\ \text{s.t.} & x_i + x_j \geq 1, & (i,j) \in E, \\ & x_i \in \{0,1\}, & i \in V. \end{array}$$

Let us denote the optimum value of this integer program, i.e., the weight of the optimal vertex cover, by $vc(G)$.

It is well known that $vc(G)$ can be approximated to within a factor of 2 in polynomial time; one way to see this is as follows. We can relax the constraint that the x_i be 0-1 variables, obtaining the following linear program:

$$\begin{array}{lll} \text{(LP)} \quad \text{Min} & \sum_i w_i x_i & \\ \text{s.t.} & x_i + x_j \geq 1, & (i,j) \in E, \\ & 0 \leq x_i \leq 1, & i \in V. \end{array}$$

*Received by the editors June 12, 1995; accepted for publication (in revised form) June 29, 1997. http://www.siam.org/journals/sidma/11-2/28754.html

†Department of Computer Science, Cornell University, Ithaca, NY 14853 (kleinber@cs.cornell.edu). The research of this author was performed at MIT and supported by an ONR Graduate Fellowship.

‡Department of Mathematics, MIT, Cambridge, MA 02139 (goemans@math.mit.edu). The research of this author was supported by NSF contract 9302476-CCR and an NEC research grant.

Let us denote the optimum value of (LP) by $lp(G)$. Then clearly $vc(G) \geq lp(G)$, but we also have that $lp(G) \geq vc(G)/2$, as the set

$$\{i : x_i \geq 1/2\},$$

in any feasible solution x to (LP) is easily seen to be a vertex cover for G (Hochbaum [7]). Thus, this linear program leads to a 2-approximation algorithm for the vertex cover problem.

There has been considerable work on the problem of finding a polynomial-time approximation algorithm with an improved performance guarantee; the best bound currently known is $2 - \frac{\log \log n}{2 \log n}$ [2, 13]. What is quite striking is that no polynomial-time $(2-\varepsilon)$-approximation algorithm is known, for any constant $\varepsilon > 0$.

1.1. The present work. In this note, we consider a number of natural semidefinite programming relaxations of the vertex cover problem and investigate whether any of these might provide a $(2-\varepsilon)$-approximation algorithm. Semidefinite programming relaxations have recently proved useful in obtaining improved approximation algorithms for a number of well-studied optimization problems, including maximum cut and satisfiability problems [6], vertex coloring [9], and the maximum independent set problem [1]. Probably the most well-known semidefinite programming relaxation is the *theta function* $\vartheta(G)$ of Lovász [11]. This was introduced as a relaxation of the maximum independent set problem and used in [11] to show the polynomial-time solvability of the maximum independent set and minimum vertex coloring problems in perfect graphs. It has been used recently in the approximation algorithms of [9] and [1].

Let $\alpha(G)$ denote the maximum weight of an independent set of G, and let $W = \sum_{i \in V} w_i$ denote the sum of all vertex weights in G. Since $vc(G) = W - \alpha(G)$, it is natural to ask how well $W - \vartheta(G)$ approximates $vc(G)$. It is not difficult to show (see section 2) that $W - \vartheta(G)$ is always at least $lp(G)$, and hence not more than a factor of 2 smaller than $vc(G)$. Our main result is a corresponding lower bound; we construct a family of unweighted graphs for which the ratio of $vc(G)$ to $n - \vartheta(G)$ converges to 2, where $n = |V|$.

The techniques involved in our construction of the lower bound have also been developed in independent work of Alon and Kahale [1] and Karger, Motwani, and Sudan [9]. In particular, the gap between $vc(G)$ and $n - \vartheta(G)$ can also be obtained from a construction due independently to Alon and Kahale [1]. Their concern was with the complement of our problem: graphs G with small independence number for which $\vartheta(G)$ converges to $\frac{1}{2}n$. We also note that the recent construction of Feige [4], showing that the ratio $\vartheta(G)/\alpha(G)$ can be as large as $n^{1-o(1)}$, is of no use for our purposes; for the graphs he deals with, the ratio of $vc(G)$ to $n - \vartheta(G)$ converges to 1, not 2.

In the final section, we present a natural strengthening of the formulation; this turns out to be equal to $W - \vartheta'(G)$, where ϑ' denotes the variant of the Lovász theta function introduced by Schrijver [14]. We currently do not know of families of graphs for which the ratio of $W - \vartheta'(G)$ to $vc(G)$ converges to 2, and we indicate how the question of the existence of such examples is closely related to some open problems in combinatorial geometry.

2. The semidefinite programming relaxation. Perhaps the most natural way to obtain our semidefinite programming relaxation is by considering the following

quadratic integer programming formulation of $vc(G)$.

$$\begin{array}{lll} \text{(VC)} \quad \text{Min} & \sum_{i\in V} w_i(1+y_0y_i)/2 & \\ \text{s.t.} & (y_0-y_i)(y_0-y_j)=0, & (i,j)\in E,\\ & y_i\in\{-1,+1\}, & i\in V,\\ & y_0\in\{-1,+1\}, & \end{array}$$

where the vertex cover corresponds to the set of vertices i for which $y_i = y_0$. One could of course get rid of y_0 and/or restrict y_i to be in $\{0,1\}$, but this form simplifies the derivation of the relaxation. We now relax this integer program to one in which y_0 and y_i $(i \in V)$ are vectors in $\mathbf{R}^{n+1}$ (where n denotes $|V|$).

$$\begin{array}{lll} \text{(SD)} \quad \text{Min} & \sum_{i\in V} w_i(1+y_0\cdot y_i)/2 & \\ \text{s.t.} & (y_0-y_i)\cdot(y_0-y_j)=0, & (i,j)\in E,\\ & y_i^2=1, & i\in V,\\ & y_0^2=1. & \end{array}$$

The constraints $(y_0 - y_i) \cdot (y_0 - y_j) = 0$ for $(i, j) \in E$ can also be expressed more geometrically by saying that the midpoint $\frac{1}{2}(y_i+y_j)$ must be on the sphere centered at $y_0/2$ and of radius $\frac{1}{2}$, i.e., that $(\frac{y_i+y_j-y_0}{2})^2 = \frac{1}{4}$. The relaxation can be reformulated as a semidefinite program and therefore, using the ellipsoid algorithm, one can determine its optimum to within additive errors in polynomial time. Let us denote the optimum value of this semidefinite program by $sd(G)$. Observe that $sd(G) \le vc(G)$, since for any vertex cover S of G, we obtain a feasible solution to the above semidefinite program as follows. Set y_0 equal to any unit vector u, and for each $i \in V$, set $y_i = y_0$ if $i \in S$ and $y_i = -y_0$ if $i \notin S$.

First let us establish that we are indeed dealing with the theta function.

THEOREM 2.1. $W - sd(G) = \vartheta(G)$.

Proof. We can write $W - sd(G)$ as

$$\begin{array}{lll} \text{(SD}^c\text{)} \quad \text{Max} & \sum_{i\in V} w_i(1-y_0\cdot y_i)/2 & \\ \text{s.t.} & (y_0-y_i)\cdot(y_0-y_j)=0, & (i,j)\in E,\\ & y_i^2=1, & i\in V,\\ & y_0^2=1. & \end{array}$$

We use the following formulation of the theta function [11]; there is a unit vector $u_i \in \mathbf{R}^{n+1}$ for each vertex of G and an additional unit vector $d \in \mathbf{R}^{n+1}$.

$$\begin{array}{lll} (\vartheta) \quad \text{Max} & \sum_{i\in V} w_i(d\cdot u_i)^2 & \\ \text{s.t.} & u_i\cdot u_j=0, & (i,j)\in E,\\ & u_i^2=1, & i\in V,\\ & d^2=1. & \end{array}$$

We claim first that $W - sd(G) \le \vartheta(G)$. Given a feasible solution to (SD^c), set $d = y_0$; for each $i \in V$, we set

$$u_i = \frac{y_0 - y_i}{\|y_0 - y_i\|}$$

if $y_0 \neq y_i$; otherwise we choose u_i to be any unit vector orthogonal to d and to all other unit vectors u_j. For this set of unit vectors, we have $u_i \cdot u_j = 0$ for $(i,j) \in E$. We compute the value of the objective function as follows. If $y_0 \neq y_i$, then

$$\begin{aligned}(d \cdot u_i)^2 &= \frac{[y_0 \cdot (y_0 - y_i)]^2}{(y_0 - y_i)^2} \\ &= \frac{(1 - y_0 \cdot y_i)^2}{2(1 - y_0 \cdot y_i)} \\ &= \frac{1}{2}(1 - y_0 \cdot y_i).\end{aligned}$$

If $y_0 = y_i$, then

$$(d \cdot u_i)^2 = 0 = \frac{1}{2}(1 - y_0 \cdot y_i).$$

As a result, we have constructed a feasible solution to (ϑ) of value $W - sd(G)$.

Conversely, we show that $\vartheta(G) \leq W - sd(G)$. Given a feasible solution to (ϑ), write $y_0 = d$ and $y_i = d - 2(d \cdot u_i)u_i$. Then $y_i^2 = 1$, and if $(i,j) \in E$, we have

$$(y_0 - y_i) \cdot (y_0 - y_j) = 4(d \cdot u_i)(d \cdot u_j)(u_i \cdot u_j) = 0.$$

Finally,

$$\frac{1}{2}(1 - y_0 \cdot y_i) = \frac{1}{2}(2(d \cdot u_i)^2) = (d \cdot u_i)^2. \qquad \square$$

The next two results determine the exact approximation ratio achieved by $sd(G)$, specifically $vc(G) \leq 2sd(G)$, but, for any $\varepsilon > 0$, there exist instances for which $vc(G) > (2 - \varepsilon)sd(G)$. It is worth noting, however, that on many natural examples, $sd(G)$ is a much tighter relaxation than $lp(G)$. For instance on K^n, the complete graph on n vertices with unit weights, one has $lp(G) = \frac{1}{2}n$, while $sd(G) = vc(G) = n - 1$.

PROPOSITION 2.2. *$sd(G) \geq lp(G)$.*

Proof. Suppose we have a feasible solution to (SD), and we write $x_i = (1 + y_0 \cdot y_i)/2$. Then we claim that $\{x_i : i \in V\}$ is a feasible solution to (LP). For clearly $0 \leq x_i \leq 1$, and if $(i,j) \in E$, then $(y_0 - y_i) \cdot (y_0 - y_j) = 0$, whence $y_0 \cdot y_i + y_0 \cdot y_j = 1 + y_i \cdot y_j$ and

$$x_i + x_j = \frac{3}{2} + \frac{1}{2}y_i \cdot y_j \geq 1,$$

as required. $\square$

THEOREM 2.3. *For each $\varepsilon > 0$ there is a graph G_ε on $n = n(\varepsilon)$ vertices, with all vertex weights equal to 1, for which $vc(G_\varepsilon)/sd(G_\varepsilon) \geq 2 - \varepsilon$.*

Proof. For a point $x \in \mathbf{R}^d$, let $x^{(i)}$ denote the ith coordinate of x. Also, let $e_1, \ldots, e_d$ denote the coordinate unit vectors in $\mathbf{R}^d$.

The idea is to construct a graph G_ε as follows. The vertices of G_ε will be the set of all $n = 2^m$ many m-bit strings of zeroes and ones, for some sufficiently large value of m, and two vertices will be joined by an edge if their Hamming distance is equal to $(1 - \gamma)m$, for some small $\gamma > 0$ depending on ε. Thus, two vertices will be joined if they are nearly antipodal under the Hamming metric. We then obtain a solution to (SD), in which all y_i $(i \in V)$ are nearly orthogonal to y_0, by mapping the y_i to the vertices of an "inscribed" hypercube in a copy of the m-dimensional unit ball. Thus

$sd(G_\varepsilon)$ is close to $n/2$. Using a theorem of Frankl and Rödl [5], we can show that G_ε does not have large independent sets and thus show that $vc(G_\varepsilon)$ is close to n.

The details are as follows. Let ε' be a rational number such that $\varepsilon' \leq \varepsilon$. Let

$$\alpha = \frac{\varepsilon'}{4},$$
$$\beta = \sqrt{1-\alpha^2},$$
$$\gamma = \frac{1}{2} - \frac{(1-\alpha)^2}{2\beta^2}.$$

Note that $\gamma > 0$. The vertex set of G_ε consists of all m-bit strings of zeroes and ones, where the value of m will be determined below; for now, we only require that $(1-\gamma)m$ be an even integer. If i and j are vertices of G_ε, then $(i,j) \in E$ iff the Hamming distance between i and j is equal to $(1-\gamma)m$.

First we compute an upper bound on $sd(G_\varepsilon)$. To do this, we construct the following unit vectors in $\mathbf{R}^{m+1}$. Set $y_0 = e_{m+1}$. For $i \in V$, define y_i so that $y_i^{(p)} = \beta/\sqrt{m}$ if the pth bit of i is 1 and $y_i^{(p)} = -\beta/\sqrt{m}$ if it is 0. Finally, $y_i^{(m+1)} = \alpha$ for all $i \in V$; thus all y_i are unit vectors.

Now, if $(i,j) \in E$, then i and j have Hamming distance $(1-\gamma)m$, and hence

$$\begin{aligned}(y_0 - y_i)\cdot(y_0 - y_j) &= (1-\alpha)^2 + \gamma m(\beta^2/m) - (1-\gamma)m(\beta^2/m)\\ &= (1-\alpha)^2 - \beta^2(1-2\gamma)\\ &= 0\end{aligned}$$

by the definition of γ. Thus the given vectors constitute a feasible solution for (SD). Moreover, the value of the objective function with these vectors is equal to $\frac{1}{2}(1+\alpha)n$, so

$$sd(G_\varepsilon) \leq \frac{1}{2}(1+\alpha)n.$$

Now we show a lower bound on $vc(G_\varepsilon)$; for this we need the following theorem of Frankl and Rödl [5].

> *Let $\mathcal{C}$ be a collection of m-bit strings, ξ a constant satisfying $0 < \xi < \frac{1}{2}$, and d an even integer satisfying $\xi m < d < (1-\xi)m$. Then for some constant δ depending only on ξ, if $|\mathcal{C}| > (2-\delta)^m$, then $\mathcal{C}$ contains two strings with Hamming distance exactly d.*

For our purposes, choose $\xi < \gamma$ and let δ denote the constant obtained by applying this theorem. Now, let $d = (1-\gamma)m$, where m is chosen large enough so that d is an even integer and

$$(2-\delta)^m \leq \alpha \cdot 2^m.$$

Thus, in G_ε any set of more than $\alpha \cdot 2^m = \alpha n$ vertices contains the two endpoints of some edge, so the largest independent set in G_ε has size at most αn. Since the complement of any vertex cover is an independent set, this implies

$$vc(G_\varepsilon) \geq (1-\alpha)n.$$

The theorem now follows, since

$$\frac{vc(G_\varepsilon)}{sd(G_\varepsilon)} \geq \frac{(1-\alpha)n}{\frac{1}{2}(1+\alpha)n}$$
$$\geq 2-\varepsilon. \qquad \square$$

3. Strengthening the relaxation. It turns out that we can add a set of very natural valid inequalities to (SD) that rules out the bad example of the previous section. As we remarked in the introduction, this new formulation (SD′) is in fact equal to $W - \vartheta'(G)$, where ϑ' denotes the variant of the Lovász theta function introduced by Schrijver [14].

The new formulation is obtained by observing the following. We saw that for any vertex cover S, we can obtain a feasible solution to (SD) by setting $y_i = y_0$ for $i \in S$ and $y_i = -y_0$ for $i \notin S$. But such a solution satisfies the conditions $(y_0 - y_i) \cdot (y_0 - y_j) \geq 0$ *for all* pairs of vertices $i, j \in V$, regardless of whether $(i,j) \in E$. Thus we can write

$$\begin{array}{lll} (\mathrm{SD}') \quad \text{Min} & \sum_{i\in V} w_i(1+y_0\cdot y_i)/2 & \\ \text{s.t.} & (y_0-y_i)\cdot(y_0-y_j)=0, & (i,j)\in E, \\ & (y_0-y_i)\cdot(y_0-y_j)\geq 0, & \forall\ i,j\in V, \\ & y_i^2=1, & i\in V, \\ & y_0^2=1. & \end{array}$$

Let us denote the optimum value of (SD′) by $sd'(G)$.

The function $\vartheta'(G)$ was introduced by Schrijver [14]. As in the definition of ϑ, we have a unit vector $u_i \in \mathbf{R}^{n+1}$ for each vertex of G and an additional unit vector $d \in \mathbf{R}^{n+1}$. We can now formulate $\vartheta'(G)$ as follows.

$$\begin{array}{lll} (\vartheta') \quad \text{Max} & \sum_{i\in V} w_i(d\cdot u_i)^2 & \\ \text{s.t.} & u_i\cdot u_j=0, & (i,j)\in E, \\ & u_i\cdot u_j\geq 0, & \forall\ i,j\in V, \\ & d\cdot u_i\geq 0, & i\in V, \\ & u_i^2=1, & i\in V, \\ & d^2=1. & \end{array}$$

By a straightforward modification of the proof of Theorem 2.1, we have the following.

THEOREM 3.1. $sd'(G) = W - \vartheta'(G)$.

Now it is easy to verify that the set of vectors we constructed in the proof of Theorem 2.3 is no longer feasible for (SD′). But in fact we can say more. Let $U = \{u_1, \ldots, u_n\}$ denote a set of points in $\mathbf{R}^d$, and define d_U by

$$d_U = \max_{u_i, u_j \in U} \|u_i - u_j\|.$$

We now associate a graph $\mathcal{K}_U$ with U as follows. $\mathcal{K}_U$ contains a vertex i for each $u_i \in U$; we join i and j by an edge iff $\|u_i - u_j\| = d_U$.

Graphs of the form $\mathcal{K}_U$ are of considerable interest in combinatorial geometry because of their role in the well-known Borsuk conjecture [3], which asked (in its

finite form) whether $\chi(\mathcal{K}_U) \leq d+1$ for all point sets U in $\mathbf{R}^d$. (This is the bound achieved, for example, by the unit d-simplex.) This was recently answered negatively by Kahn and Kalai [8], who constructed, for infinitely many values of d, a set U in $\mathbf{R}^d$ for which $\chi(\mathcal{K}_U) \geq (1.2)^{\sqrt{d}}$.

Here we ask a related question. Let S^{d-1} denote the unit sphere centered at the origin in $\mathbf{R}^d$.

QUESTION 1. *Do there exist absolute constants $\varepsilon > 0$ and $\delta > 0$ so that, for all sets U of n points on S^{d-1}, $d_U \geq 2-\varepsilon$ implies $\alpha(\mathcal{K}_U) \geq \delta n$?*

That is, does every point set of sufficiently large diameter on S^{d-1} have a linear-sized independent set in its graph $\mathcal{K}_U$? It is important to note that the constants ε and δ do not depend on n or d.

The relation of this to our formulation (SD′) is contained in the following fact.

PROPOSITION 3.2. *If for some $c < 2$ we have $vc(G)/sd'(G) < c$ for all graphs G, then Question 1 has an affirmative answer.*

Proof. Given $c < 2$, let $\varepsilon = 1 - \frac{c}{2} > 0$ and $\delta = \varepsilon^2/4 > 0$. Consider a set $U = \{u_1, \ldots, u_n\}$ on S^{d-1} for which $d_U \geq 2-\varepsilon$.

We first claim that $sd'(\mathcal{K}_U) \leq \frac{2}{(2-\varepsilon)^2} n$. Select any unit vector y_0 orthogonal to all u_i's (adding one dimension if necessary). Let $y_i = \beta y_0 + \sqrt{1-\beta^2}u_i$, where $\beta = \frac{4}{d_U^2} - 1$. Observe that y_i is a unit vector and that

$$\begin{aligned}(y_0 - y_i)\cdot(y_0 - y_j) &= ((1-\beta)y_0 - \sqrt{1-\beta^2}u_i)\cdot((1-\beta)y_0 - \sqrt{1-\beta^2}u_j) \\ &= (1-\beta)^2 + (1-\beta^2)(u_i \cdot u_j) \\ &= (1-\beta)\left(1-\beta+(1+\beta)u_i\cdot u_j\right). \end{aligned} \tag{1}$$

Since the u_i are unit vectors, $||u_i - u_j||^2 = 2 - 2u_i \cdot u_j$. Substituting this into (1) we derive that

$$\begin{aligned}(y_0 - y_i)\cdot(y_0 - y_j) &= (1-\beta)\left(2 - \frac{1+\beta}{2}||u_i - u_j||^2\right) \\ &\geq (1-\beta)\left(2 - \frac{1+\beta}{2}d_U^2\right) \\ &= 0,\end{aligned}$$

with equality if $||u_i - u_j|| = d_U$. We have therefore constructed a feasible solution to (SD') of value $\frac{1+\beta}{2}n = \frac{2}{d_U^2}n \leq \frac{2}{(2-\varepsilon)^2}n$,

$$vc(\mathcal{K}_U) < c\frac{2}{(2-\varepsilon)^2}n = \frac{4-4\varepsilon}{4-4\varepsilon+\varepsilon^2}n < \left(1 - \frac{\varepsilon^2}{4}\right)n = (1-\delta)n,$$

implying that $\alpha(\mathcal{K}_U) > \delta n$. □

We do not know the answer to Question 1, but it is worth remarking on its relation to a number of other questions.

3.1. The counterexample to Borsuk's conjecture. In their counterexample to Borsuk's conjecture, Kahn and Kalai construct a family of sets $\{U_n\}$ such that U_n is a set of n points on a unit sphere, and $\alpha(\mathcal{K}_U) = o(n)$. However, their sets U_n also have $d_{U_n} = \sqrt{2} + o(1)$. Thus, following the proof of Proposition 3.2, their construction is not sufficient to exhibit a family of graphs $\{G_n\}$ for which $vc(G_n)/sd'(G_n) \geq c'$, for *any* constant $c' > 1$.

3.2. Strongly self-dual polytopes. In [12], Lovász introduces the notion of a *strongly self-dual polytope*, which is defined as a polytope P in $\mathbf{R}^d$ with the following properties:

(i) The vertices of P all lie on S^{d-1}.

(ii) For some $0 < r < 1$, P is circumscribed around the sphere of radius r centered at the origin.

(iii) There is a bijection between the vertices and facets of P so that the vector from the origin to any vertex of P is orthogonal to the corresponding facet.

Note that in $\mathbf{R}^2$, an odd regular polygon satisfies these conditions. Let $U(P)$ denote the vertices of P. Lovász proves the following two facts:

- For each dimension d, and all $\varepsilon > 0$, there is a strongly self-dual polytope P in $\mathbf{R}^d$ with $d_{U(P)} \geq 2 - \varepsilon$.
- For any strongly self-dual polytope P in $\mathbf{R}^d$, $\chi(\mathcal{K}_{U(P)}) \geq d + 1$.

Taken together, these two facts provide a negative answer to the following—a slight modification of a question due to Erdős and Graham.

QUESTION 2. *Do there exist absolute constants $\varepsilon > 0$ and $C > 0$ so that, for all sets U of n points on S^{d-1}, $d_U \geq 2 - \varepsilon$ implies $\chi(\mathcal{K}_U) \leq C$?*

Clearly, an affirmative answer to Question 2 would have implied an affirmative answer to Question 1. On the other hand, an affirmative answer to Question 1 implies that if $d_U \geq 2 - \varepsilon$, then $\chi(\mathcal{K}_U) \leq 1 + \log_{(1/(1-\delta))} n = O(\log n)$. This simply corresponds to repeatedly coloring a fraction δ of the vertex set with a new color. For this argument, we also need to observe that we could apply the affirmative answer to Question 1 to an induced subgraph of $\mathcal{K}_U$ (unless it is itself an independent set).

However, looking at Lovász's construction, one finds that for the strongly self-dual polytopes P he constructs, one always has $\alpha(\mathcal{K}_{U(P)}) \geq \frac{n-1}{2}$ (with equality achieved, for example, on all regular odd polygons in $\mathbf{R}^2$). Thus, his construction is not able to provide a negative answer to Question 1; as with [8], it does not provide a family of graphs $\{G_n\}$ for which $vc(G_n)/sd'(G_n) \geq c'$, for any constant $c' > 1$.

Naturally, it would be interesting to investigate other constructions of strongly self-dual polytopes.

3.3. A graph-coloring formulation. Consider the following special case of Question 1, to which we also do not know the answer.

QUESTION 3. *Can one take some $\varepsilon > 2 - \sqrt{3}$ in Question 1?*

Then we have the following proposition.

PROPOSITION 3.3. *If Question 3 has an affirmative answer, then for some $C > 0$ one can prove in polynomial time that a graph of chromatic number at least $C \log n$ is not 3-colorable.*

Proof. Given a graph $G = (V, E)$, consider the problem of finding unit vectors u_i such that E is a subset of the edge set of $\mathcal{K}_U$ and such that d_U is maximized. Let d_{max} be the maximum achievable. Since $||u_i - u_j||^2 = 2 - 2u_i \cdot u_j$, this problem can be formulated in terms of a semidefinite program

$$\begin{array}{lll} \text{Min} & z & \\ \text{s.t.} & u_i \cdot u_j = z, & (i,j) \in E, \\ & u_i \cdot u_j \geq z, & (i,j) \notin E, \\ & u_i^2 = 1, & i \in V. \end{array}$$

Now suppose that Question 1 has an affirmative answer for *some* constants $\varepsilon > 0$ and $\delta > 0$. By the remark following Question 2, we know that if the chromatic number of G is at least $C \log n$ (for some C depending on δ), then there are no vectors u_i with the

desired properties for which $d_U \geq 2 - \varepsilon$. But, in polynomial time, we can determine d_{max} to within any additive error by solving the above semidefinite program. In particular, if $\varepsilon > 2 - \sqrt{3}$, we can prove in polynomial time that $d_{max} < \sqrt{3}$. This constitutes a proof that the graph is not 3-colorable, since a 3-coloring would imply $d_{max} \geq \sqrt{3}$ (as in [9]). □

Note added in proof. Jens Lagergren and Alexander Russell have recently announced that, by weighting Lovász's construction of strongly self-dual polytopes, one can obtain weighted graphs for which $vc(G)/sd'(G)$ is arbitrarily close to 2 [10].

REFERENCES

[1] N. Alon and N. Kahale, *Approximating the independence number via the θ-function*, manuscript, 1996.

[2] R. Bar-Yehuda and S. Even, *A local-ratio theorem for approximating the weighted vertex cover problem*, in Analysis and Design of Algorithms for Combinatorial Problems, Elsevier, Amsterdam, 1985, pp. 27–46.

[3] K. Borsuk, *Drei Sätze über die n-dimensionale euklidische Sphäre*, Fundamenta Math., 20 (1933), pp. 177–190.

[4] U. Feige, *Randomized graph products, chromatic numbers, and the Lovász θ-function*, in Proc. 27th Annual ACM Symposium on Theory of Computing, ACM, New York, 1995, pp. 635–640.

[5] P. Frankl and V. Rödl, *Forbidden intersections*, Trans. Amer. Math. Soc., 300 (1987), pp. 259–286.

[6] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42 (1995), pp. 1115–1145. (Preliminary version in Proc. 26th Annual ACM Symposium on Theory of Computing, ACM, New York, 1994, pp. 422–431.)

[7] D. Hochbaum, *Approximation algorithms for set covering and vertex cover problems*, SIAM J. Comput., 11 (1982), pp. 555–556.

[8] J. Kahn and G. Kalai, *A counterexample to Borsuk's conjecture*, Bull. Amer. Math. Soc., 29 (1993), pp. 60–62.

[9] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, in Proc. 35th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 2–13.

[10] J. Lagergren and A. Russell, *Vertex Cover Eludes the Schrijver Function*, manuscript, 1997.

[11] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[12] L. Lovász, *Self-dual polytopes and the chromatic number of distance graphs on the sphere*, Acta Sci. Math., 45 (1983), pp. 317–323.

[13] B. Monien and E. Speckenmeyer, *Ramsey numbers and an approximation algorithm for the vertex cover problem*, Acta Inform., 22 (1985), pp. 115–123.

[14] A. Schrijver, *A comparison of the Delsarte and Lovász bounds*, IEEE Trans. Inform. Theory, 25 (1979), pp. 425–429.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 205–223, May 1998

003

ON PERFECT CODES AND TILINGS: PROBLEMS AND SOLUTIONS*

TUVI ETZION† AND ALEXANDER VARDY‡

Abstract. Although nontrivial perfect binary codes exist only for length $n = 2^m - 1$ with $m \geq 3$ and for length $n = 23$, many problems concerning these codes remain unsolved. Herein, we present solutions to some of these problems. In particular, we show that the smallest nonempty intersection of two perfect codes of length $2^m - 1$ consists of two codewords, for all $m \geq 3$. We also provide a complete solution to the intersection number problem for Hamming codes. Furthermore, we prove that a perfect code of length $2^{m-1} - 1$ is embedded in a perfect code $\mathbb{C}$ of length $2^m - 1$ if and only if $\mathbb{C}$ is not of full rank. This result implies the existence of distinct generalized Hamming weights for perfect codes, and we determine completely the generalized Hamming weights of all perfect codes that do not contain embedded full-rank perfect codes. We further explore the close ties between perfect codes and tilings: we prove that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 14$ and show that the existence of full-rank tilings for other n is closely related to the existence of full-rank perfect codes with kernels of high dimension. We briefly survey the present state of knowledge on perfect binary codes and list several interesting and important open problems concerning perfect codes and tilings.

Key words. perfect codes, tilings, intersection, embedding, rank, kernel

AMS subject classifications. O5A18, O5B40, 20K01, 94B25, 94B60

PII. S0895480196309171

1. Introduction.

Let $\mathbb{F}_2^n$ be a vector space of dimension n over GF(2). A subset of $\mathbb{F}_2^n$ is a binary code of length n. Two codes $\mathbb{C}_1, \mathbb{C}_2 \subset \mathbb{F}_2^n$ are *isomorphic* if there exists a permutation π such that $\mathbb{C}_2 = \pi(\mathbb{C}_1) = \{\pi(c) : c \in \mathbb{C}_1\}$. They are *equivalent* if there exists a vector a and a permutation π such that $\mathbb{C}_2 = a + \pi(\mathbb{C}_1) = \{a + \pi(c) : c \in \mathbb{C}_1\}$. The Hamming *distance* between vectors $x, y \in \mathbb{F}_2^n$, denoted $d(x, y)$, is the number of coordinates in which x and y differ. The Hamming *weight* of x is given by $\mathrm{wt}(x) = d(x, \mathbf{0})$, where $\mathbf{0}$ is the all-zero vector. Without loss of generality, we shall assume (unless stated otherwise) that $\mathbf{0} \in \mathbb{C}$, throughout this paper. We let $\langle \mathbb{C} \rangle$ denote the linear span of a code $\mathbb{C} \subset \mathbb{F}_2^n$. The rank of $\mathbb{C}$, denoted $\mathsf{rank}(\mathbb{C})$, is the dimension of $\langle \mathbb{C} \rangle$. We say that $\mathbb{C}$ is of full-rank if $\mathsf{rank}(\mathbb{C}) = n$, or equivalently, if $\langle \mathbb{C} \rangle = \mathbb{F}_2^n$.

A binary code $\mathbb{C}$ of length n is *perfect* if, for some integer $r \geq 0$, every $x \in \mathbb{F}_2^n$ is within distance r from exactly one codeword of $\mathbb{C}$. The study of perfect codes has always been one of the most fascinating subjects in coding theory. It is shown in [32, 33, 38] that such codes exist only for $r = 0$, $r = n$, $r = (n-1)/2$ with n odd, $r = 1$ with $n = 2^m - 1$, and $r = 3$ with $n = 23$. The first three cases are trivial, while the last case corresponds to the well-known binary Golay code [20], which is known to be unique up to equivalence [7, 28, 29]. Thus the only parameters for which there

*Received by the editors September 10, 1996; accepted for publication (in revised form) May 8, 1997. This work was partially supported by grant 95-522 from the United States–Israel Binational Science Foundation.

http://www.siam.org/journals/sidma/11-2/30917.html

†Department of Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel (etzion@cs.technion.ac.il). The research of this author was supported in part by the fund for promotion of sponsored research at the Technion.

‡Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801 (vardy@golay.csl.uiuc.edu). The research of this author was supported by the David and Lucile Packard Foundation, the National Science Foundation, and JSEP grant N00014–9610129.

exist inequivalent perfect binary codes are $r = 1$ and $n = 2^m - 1$, with $m \geq 4$, and we shall henceforth use the word "perfect" to refer specifically to codes of this type. The linear perfect codes are, again, unique up to equivalence — these are the well-known Hamming codes [20]. Nonlinear perfect codes were constructed and studied in [1, 5, 9, 12, 21, 23, 24, 26, 30, 35], among other works. Some of these constructions are outlined in the next section.

Although perfect binary codes have been the subject of much research, many interesting questions regarding these codes remain open. Herein, we provide answers to some of these questions. In section 3, we answer the question raised in [9] and show that, for each $m \geq 3$, there exist two perfect codes $\mathbb{C}_1, \mathbb{C}_2$ of length $n = 2^m - 1$ such that $|\mathbb{C}_1 \cap \mathbb{C}_2| = 2$. We also consider the more general problem of the *intersection numbers* of perfect codes and provide a complete solution to this problem for the linear (Hamming) perfect codes. In section 4, we consider the problem of embedding a perfect code $\mathbb{C}_1$ of length n_1 within a perfect code $\mathbb{C}_2$ of length $n_2 > n_1$. We prove that a perfect code of length $2^{m-1} - 1$ is embedded within a perfect code $\mathbb{C}$ of length $2^m - 1$ if and only if $\mathbb{C}$ is not of full rank. This result implies that perfect codes of the same length can have distinct generalized Hamming weights (cf. [36]) and distinct cardinality-length profiles (cf. [17]). We prove that the generalized Hamming weights of a perfect code $\mathbb{C}$ coincide with those of the Hamming code of the same length, provided there is no full-rank perfect code embedded in $\mathbb{C}$. In section 5, we investigate the connections between perfect codes and tilings, answering some of the questions that were left open in [5]. It was shown in [5] that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 112$, and we prove in section 5 that, in fact, such tilings exist for all $n \geq 14$. Since full-rank tilings do not exist for $n \leq 7$ (cf. [5]), this leaves only six values of n unresolved. We show that the existence of full-rank tilings for these n is closely related to the existence of full-rank perfect codes with high-dimensional kernels. Finally, we conclude in section 6 with a list of open problems concerning perfect codes and tilings.

2. Constructions and properties of perfect codes. In this section, we briefly outline two constructions of perfect codes, termed Construction A and Construction B. These constructions will be used later in this paper. We also review some of the properties of these constructions, as well as certain properties of perfect codes in general, that are of relevance to our work.

We say that a code is *even* if all of its codewords have even weight. Given a code $\mathbb{C} \subset \mathbb{F}_2^n$ which is not even, we can extend it by an even parity coordinate to obtain an even code, called the *extended code* of $\mathbb{C}$. An even code $\mathbb{C}^*$ of length $n + 1 = 2^m$ is said to be *extended perfect* if it can be obtained by means of extending a perfect code of length n by an even parity coordinate. Notice that deleting any coordinate of an extended perfect code produces a perfect code. Also observe that Constructions A and B, described in what follows in the context of perfect codes, can be straightforwardly modified to produce extended perfect codes.

Let $\mathbb{E}_2^n$ denote the set of all the even-weight vectors in $\mathbb{F}_2^n$. For a code $\mathbb{C} \subset \mathbb{F}_2^n$ and a vector $a \in \mathbb{F}_2^n$, the code $a + \mathbb{C} = \{a + c : c \in \mathbb{C}\}$ is called a *translate* of $\mathbb{C}$. If $\mathbb{C}$ is linear, then a translate of $\mathbb{C}$ is also called a *coset*. Let e_i denote a vector of weight one with the nonzero entry in the ith position. It is easy to see that, for a perfect code $\mathbb{C} \subset \mathbb{F}_2^n$, the translates $\mathbb{C}, e_1 + \mathbb{C}, \ldots, e_n + \mathbb{C}$ form a partition of $\mathbb{F}_2^n$. Similarly, the set $\mathbb{E}_2^{n+1}$ can always be partitioned into even translates of an extended perfect code $\mathbb{C}^* \subset \mathbb{E}_2^{n+1}$. The following construction of perfect codes of length $2n + 1$ from perfect codes of length n is due to Phelps [23] and Solov'eva [30].

CONSTRUCTION A. *Let* $\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_n$ *and* $\mathbb{C}_0^*, \mathbb{C}_1^*, \ldots, \mathbb{C}_n^*$ *be partitions of* $\mathbb{F}_2^n$ *and* $\mathbb{E}_2^{n+1}$*, into a perfect code and its translates, respectively, into an extended perfect code and its translates. Let* π *be a permutation on the set* $\{0, 1, \ldots, n\}$*. Then the code*

$$\mathbb{C}_{\mathrm{A}} \;=\; \{\, (x|\, y) \;:\; x \in \mathbb{C}_i, \quad y \in \mathbb{C}^*_{\pi(i)} \quad \text{for some } i = 0, 1, \ldots, n \,\},$$

where $(\cdot|\cdot)$ *denotes concatenation, is a perfect code of length* $2^{m+1} - 1$.

We say that a vector $a \in \mathbb{F}_2^n$ *covers* a subset $\mathcal{S}_a \subset \mathbb{F}_2^n$ if $\mathcal{S}_a = \{x : d(x, a) \le 1\}$. Similarly, we say that a code $\mathbb{C} \subset \mathbb{F}_2^n$ covers a subset $\mathcal{S}$ if

$$\mathcal{S} \;=\; \{\, x \;:\; \exists\, c \in \mathbb{C} \text{ such that } d(x, c) \le 1 \,\} \;=\; \cup_{c \in \mathbb{C}}\, \mathcal{S}_c.$$

We say that $\mathbb{C}$ *perfectly covers* $\mathcal{S}$ if $\mathbb{C}$ covers $\mathcal{S}$ and $d(\mathbb{C}) \stackrel{\text{def}}{=} \min_{x,y \in \mathbb{C}} d(x, y) = 3$. The following construction of perfect codes may be found in [9]; it can be viewed as a certain special case of the construction of Vasil'ev [35]. This construction leads to perfect codes with various useful properties, such as full-rank perfect codes or perfect codes with large intersections. First, we define the following codes:

$$\begin{aligned} \mathcal{A} &= \{\, (\, x \,|\, p(x) \,|\, x\,) \;:\; x \in \mathbb{F}_2^n \,\}, \\ \mathcal{B} &= \{\, (\, x \,|\, p(x)+1 \,|\, x\,) \;:\; x \in \mathbb{F}_2^n \,\}, \end{aligned} \tag{2.1}$$

where $p(x) = \mathrm{wt}(x) \bmod 2$ is the *parity* of x. The following lemma was established in Etzion and Vardy [9].

LEMMA 2.1. *The codes* $\mathcal{A}$ *and* $\mathcal{B}$ *perfectly cover the same subset of* $\mathbb{F}_2^{2n+1}$.

Lemma 2.1 will be used in the next section to construct perfect codes with small intersection.

CONSTRUCTION B. *Assume that* $\mathbb{C}_1$ *is a perfect code of the form* $\mathbb{C}_1 = \mathbb{C}' \cup (x + \mathcal{A})$*, where* $x = \mathbf{0}$ *or* $x \notin \mathcal{A}$*. Then the code* $\mathbb{C}_2 = \mathbb{C}' \cup (x + \mathcal{B})$ *is also a perfect code.*

Construction B follows immediately from Lemma 2.1. It is shown in [9] that the set $\mathcal{A}$ is a linear subcode of the Hamming code (in an appropriate permutation). Thus one can use Construction B to produce nonlinear perfect codes from the Hamming code. In [9], we have applied this construction m times to produce a *full-rank* perfect code of length $2^m - 1$ from the Hamming code of the same length, for all $m \ge 4$. A similar approach was subsequently used by Phelps and LeVan [26] to construct perfect codes with kernels of various dimensions, while generalizations to nonbinary perfect codes were developed in [8].

Another application of Construction B enables one to construct a large set of inequivalent perfect codes. Let $\mathcal{H}_m$ denote the Hamming code of length $n = 2^m - 1$, and let $c_1, c_2, \ldots, c_t$ be the coset representatives for $\mathcal{A}$ in $\mathcal{H}_m$, where $t = 2^{0.5(n+1)-m}$. By Lemma 2.1, the sets $c_i + \mathcal{A}$ and $c_i + \mathcal{B}$ perfectly cover the same subset of $\mathbb{F}_2^n$ for all i. Thus, we have the following.

THEOREM 2.2. *To each binary vector* $x = (x_1, x_2, \ldots, x_t)$*, there corresponds a perfect code*

$$\mathbb{C}_{\langle x \rangle} \;=\; \bigcup_{i=1}^{t} \big(c_i + x_i \mathcal{A} + \bar{x}_i \mathcal{B}\big),$$

where the notation $x_i\mathcal{A} + \bar{x}_i\mathcal{B}$ *stands for either* $\mathcal{A}$ *if* $x_i = 1$ *or* $\mathcal{B}$ *if* $x_i = 0$.

Theorem 2.2 produces a set of $2^t = 2^{2^{0.5(n+1)-\log(n+1)}}$ distinct perfect codes. It was shown in [9] that the number of inequivalent perfect codes in this set is close to $2^{2^{0.5n}}$ for large n.

The weight distribution of a perfect code is uniquely determined [20, p. 129] by its length n. An explicit closed-form expression for the weight distribution of perfect codes may be found in [9]. In particular, it is known that any perfect code $\mathbb{C}$ of length n that contains $\mathbf{0}$ also contains $\mathbf{1}$ — the unique binary vector of weight n. It follows that $\mathbf{1}$ belongs to $c+\mathbb{C}$ for all $c\in\mathbb{C}$. This, in turn, implies that if $c\in\mathbb{C}$, then also $\bar{c}\in\mathbb{C}$, where $\bar{c}=\mathbf{1}+c$ is the binary complement of c. Codes with this property are called *self-complementary*, and the foregoing observation shows that all perfect codes are self-complementary.

3. Intersections of perfect codes. Given two binary codes $\mathbb{C}_1,\mathbb{C}_2$ of the same length, the *intersection number* of $\mathbb{C}_1$ and $\mathbb{C}_2$ is defined as $\eta(\mathbb{C}_1,\mathbb{C}_2)\stackrel{\text{def}}{=}|\mathbb{C}_1\cap\mathbb{C}_2|$. In this section, we consider the following problem: what are the possible intersection numbers of perfect codes of a given length? The *largest* possible intersection number of perfect codes was determined in [9]. Specifically, it was shown in [9] that if $\mathbb{C}_1,\mathbb{C}_2$ are two distinct perfect codes of length $n=2^m-1$, then

$$\eta(\mathbb{C}_1,\mathbb{C}_2)\le 2^{2^m-m-1}-2^{2^{m-1}-1}$$

and this bound is tight; namely, for all $m\ge 3$ there exist perfect codes $\mathbb{C}_1,\mathbb{C}_2$ of length 2^m-1 such that $\eta(\mathbb{C}_1,\mathbb{C}_2)=2^{2^m-m-1}-2^{2^{m-1}-1}$.

A natural counterpart to this question is: what is the *smallest* possible (nonzero) intersection number of two perfect codes? It was shown [9] that for all $m\ge 3$ there exist two perfect codes $\mathbb{C}_1,\mathbb{C}_2$ of length 2^m-1 such that

$$\eta(\mathbb{C}_1,\mathbb{C}_2)=2^{2^{m-2}}.$$

However, the question of whether this intersection number is the smallest possible was left open in [9]. In fact, as will be shown in this section, it is not. Since all perfect codes are self-complementary, their intersection must have even cardinality. This implies that if $\mathbb{C}_1,\mathbb{C}_2$ are perfect codes and $\eta(\mathbb{C}_1,\mathbb{C}_2)\ne 0$, then $\eta(\mathbb{C}_1,\mathbb{C}_2)\ge 2$. In what follows, we prove that, for each $m\ge 3$, there exist two perfect codes $\mathbb{C}_1,\mathbb{C}_2$ of length 2^m-1 such that $\eta(\mathbb{C}_1,\mathbb{C}_2)=2$.

First, it is obvious that the intersection problem, in general, has the same answer for perfect codes and for extended perfect codes. We will use this simple fact later in the paper; we therefore state it formally as the following lemma.

LEMMA 3.1. *Perfect codes of length 2^m-1 with intersection number q exist if and only if there exist extended perfect codes of length 2^m with intersection number q.*

We now use a combination of Constructions A and B of the foregoing section to construct two extended perfect codes with intersection number 2. Let $\mathcal{H}_0$ be an extended Hamming code of length 2^m, and let $\mathcal{H}_1,\mathcal{H}_2,\ldots,\mathcal{H}_{2^m-1}$ be the even cosets of $\mathcal{H}_0$ in $\mathbb{E}_2^{2^m}$. Thus $\mathcal{H}_0,\mathcal{H}_1,\mathcal{H}_2,\ldots,\mathcal{H}_{2^m-1}$ is a partition of $\mathbb{E}_2^{2^m}$ into extended perfect codes. Hence, the code

$$\mathbb{C}\;=\;\{\,(x|\,y)\;:\;x,y\in\mathcal{H}_i\quad\text{for some } i=0,1,\ldots,2^m-1\,\} \tag{3.1}$$

is an extended perfect code of length 2^{m+1} obtained through Construction A, with π being the identity permutation. Furthermore, it can be easily verified that $\mathbb{C}$ is a linear code, and hence it must be an extended Hamming code of length 2^{m+1}. Without loss

of generality, we can assume that the parity-check matrix of $\mathbb{C}$ is given by

$$H_{m+1} \;=\; \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{3.2}$$

That is, the columns of H_{m+1} are all of the $(m+1)$-tuples that end with a 1, ordered lexicographically. Indeed, it is easy to see that

$$H_{m+1} \;=\; \left[\begin{array}{c|c} 0 \cdots 0 & 1 \cdots 1 \\ \hline H_m & H_m \end{array}\right],$$

where H_m is a parity-check matrix for an extended Hamming code of length 2^m, which we take as $\mathcal{H}_0$. Thus the code defined by the parity-check matrix H_{m+1} is a Construction A perfect code consistent with (3.1). Notice that all the vectors in a given coset of $\mathcal{H}_0$ have the same syndrome with respect to H_m. That is, for all $i = 0, 1, \ldots, 2^m-1$, we have $s_i = H_m x^t$ for all $x \in \mathcal{H}_i$, and we say that s_i is the syndrome of $\mathcal{H}_i$.

We now use Construction B to modify the Hamming code $\mathbb{C}$ in (3.1) in an appropriate manner. Let

$$\mathcal{A}^* \;=\; \{\, (x|x) \;:\; x \in \mathcal{H}_i \quad \text{for some } i = 0, 1, \ldots, 2^m-1 \,\}. \tag{3.3}$$

Comparing (3.1) and (3.3), we see that $\mathcal{A}^*$ is a subcode of $\mathbb{C}$. Furthermore, since the codes $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_{2^m-1}$ form a partition of $\mathbb{E}_2^{2^m}$, we can write

$$\mathcal{A}^* = \{\, (x|x) \;:\; x \in \mathbb{E}_2^{2^m} \,\},$$

which implies that $\mathcal{A}^*$ is just the extended code of $\mathcal{A}$ in (2.1). Pick a fixed integer j in the range $1 \le j \le 2^m$, and let $\mathcal{B}^* = (e_j|e_j) + \mathcal{A}^*$. Then $\mathcal{B}^*$ is the extended code of $\mathcal{B}$ in (2.1). This implies that the code

$$\mathbb{C}' \;=\; \big(\mathbb{C} \setminus \mathcal{A}^*\big) \cup \mathcal{B}^*$$

is an extended perfect code, obtained by Construction B. We note that $\mathbb{C}'$ does not contain the all-zero vector; however, the translate $\mathbb{C}_1 = (e_j|e_j) + \mathbb{C}'$ does. This translate is an extended perfect code, which can be written as $\mathbb{C}_1 = \mathcal{A}^* \cup \mathcal{D}$, where

$$\mathcal{D} \;=\; \{\, (x + e_j \,|\, y + e_j) \;:\; x, y \in \mathcal{H}_i \text{ and } x \ne y, \quad \text{for some } i = 0, 1, \ldots, 2^m-1 \,\}.$$

Now, let π be the permutation that fixes the last 2^m coordinates of $\mathbb{C}_1$ and effects the cyclic shift by one position on the first 2^m coordinates. Define $\mathbb{C}_2 = \pi(\mathbb{C}_1)$. Then obviously $\mathbb{C}_2$ is a perfect code, and we have the following.

THEOREM 3.2. *The intersection number of* $\mathbb{C}_1$ *and* $\mathbb{C}_2$ *is* $\eta(\mathbb{C}_1, \mathbb{C}_2) = 2$.

Proof. Suppose $(x|\,y) \in \mathbb{C}_1$, for some $x, y \in \mathbb{F}_2^{2^m}$. Then clearly $\mathrm{wt}(x) \equiv \mathrm{wt}(y) \equiv 0$ modulo 2 if and only if $x = y$ and $(x|\,y) \in \mathcal{A}^*$, while $\mathrm{wt}(x) \equiv \mathrm{wt}(y) \equiv 1 \bmod 2$ if and only if $(x|\,y) \in \mathcal{D}$. Since the permutation π preserves the weight of x and y, we have

$$\mathbb{C}_1 \cap \mathbb{C}_2 \;=\; \big(\mathcal{A}^* \cap \pi(\mathcal{A}^*)\big) \;\cup\; \big(\mathcal{D} \cap \pi(\mathcal{D})\big). \tag{3.4}$$

A vector $x \in \mathbb{F}_2^{2^m}$ is equal to its own cyclic shift by one position if and only if $x \in \{\mathbf{0}, \mathbf{1}\}$. Hence $\mathcal{A}^* \cap \pi(\mathcal{A}^*) = \{\mathbf{0}, \mathbf{1}\}$. We now show that $\mathcal{D} \cap \pi(\mathcal{D}) = \varnothing$. First, notice that for each $(x|\, y) \in \mathcal{D}$, we have

$$H_m x^t \;=\; H_m y^t \;=\; s_i + H_m(e_j)^t \tag{3.5}$$

for some $i = 0, 1, \ldots, 2^m - 1$. On the other hand, it can be shown that if $(x|\, y) \in \pi(\mathcal{D})$, then $H_m x^t \neq H_m y^t$. Indeed, let $(x'|\, y') \in \mathcal{D}$ be the preimage of $(x|\, y)$ under π. That is, $y = y'$ and x is the cyclic shift of x' by one position. Then $H_m y^t = H_m(y')^t = H_m(x')^t$ by (3.5). Now, both x' and its cyclic shift x have odd weight, and therefore

$$(0101\cdots01)(x')^t \;\neq\; (0101\cdots01)\, x^t.$$

Since $(0101\cdots01)$ is a row of H_m, it follows that $H_m x^t \neq H_m(x')^t = H_m(y)^t$. Comparing this with (3.5), we conclude that $\mathcal{D} \cap \pi(\mathcal{D}) = \varnothing$. In conjunction with (3.4), this implies that $\mathbb{C}_1 \cap \mathbb{C}_2 = \mathcal{A}^* \cap \pi(\mathcal{A}^*) = \{\mathbf{0}, \mathbf{1}\}$, and therefore $\eta(\mathbb{C}_1, \mathbb{C}_2) = 2$. □

It follows from Theorem 3.2 and the results of [9] that the intersection number of any two distinct perfect codes $\mathbb{C}_1, \mathbb{C}_2$ of length $n = 2^m - 1$ is in the range

$$2 \;\leq\; \eta(\mathbb{C}_1, \mathbb{C}_2) \;\leq\; 2^{2^m - m - 1} - 2^{2^{m-1} - 1}, \tag{3.6}$$

and both bounds are achievable for all $m \geq 3$. Since perfect codes are self-complementary, their intersection numbers must be even. Thus a natural question is: which even integers in the range of (3.6) are intersection numbers of perfect codes of length $2^m - 1$? Using Theorem 2.2 of the previous section, we obtain intersection numbers of the form

$$k\, 2^{2^{m-1} - 1} \quad \text{for all} \quad k = 1, 2, \ldots, 2^{2^{m-1} - m} - 1.$$

These correspond to the intersection of $\mathbb{C}_{\langle \mathbf{0} \rangle}$ with $\mathbb{C}_{\langle x \rangle}$, where x is a binary vector of length $t = 2^{2^{m-1} - m}$ and weight $t - k$. Further, using modifications of Constructions A and B, along with the techniques developed in this section, we can obtain many more intersection numbers. In general, however, the problem of enumerating all possible intersection numbers of perfect codes remains open. By and large, this appears to be a difficult problem. For perfect codes of length 15, we have generated a large set of intersection numbers through a combination of known constructions and computer search. Even for this case, however, complete enumeration does not seem to be within easy reach.

A variant of the problem discussed in the previous paragraph asks for all possible intersection numbers of *linear* perfect codes, namely, the Hamming codes of length $2^m - 1$. In what follows, we provide a complete solution to this problem.

Let $\mathcal{H}_1, \mathcal{H}_2$ be two Hamming codes of length $n = 2^m - 1$. Since Hamming codes are unique, $\mathcal{H}_1$ and $\mathcal{H}_2$ are necessarily isomorphic. Since both codes are linear, their intersection number is necessarily a power of 2. For $m = 3$ and $n = 7$, it is easy to find specific permutations such that $\eta(\mathcal{H}_1, \mathcal{H}_2) = 2, 4$, or 8. For example, let $\mathcal{H}_1$ be a code defined by the parity-check matrix whose columns are ordered lexicographically, and let $\mathcal{H}_2$ be a code defined by the parity-check matrix

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

respectively. We will show that a similar situation occurs for all $m \geq 3$, namely, all the powers of 2 in the range $2^{n-2m}, 2^{n-2m+1}, \ldots, 2^{n-m-1}$ are attainable as intersection numbers of distinct Hamming codes of length $n = 2^m - 1$.

Let H_1, H_2 be parity-check matrices of the Hamming codes $\mathcal{H}_1$ and $\mathcal{H}_2$ of length $n = 2^m - 1$. Then $\mathbb{C} = \mathcal{H}_1 \cap \mathcal{H}_2$ is a linear code, whose parity-check matrix is given by

$$H \;=\; \left[\begin{array}{c} H_1 \\ \hline H_2 \end{array}\right]. \tag{3.8}$$

For the sake of brevity, we shall henceforth write $H = H_1 \| H_2$ to denote the structure of (3.8). It is obvious that $\mathsf{rank}(H) \le 2m$, since H_1 and H_2 each have m rows, and therefore,

$$\eta(\mathcal{H}_1, \mathcal{H}_2) \;=\; |\mathbb{C}| \;=\; 2^{n-\mathsf{rank}(H)} \;\ge\; 2^{n-2m}.$$

It is also obvious that $\eta(\mathcal{H}_1, \mathcal{H}_2) \le 2^{n-m-1}$ if the codes $\mathcal{H}_1$ and $\mathcal{H}_2$ are distinct.

LEMMA 3.3. *For each $m \ge 3$, there exist two Hamming codes $\mathcal{H}_1, \mathcal{H}_2$ of length $n = 2^m - 1$ such that $\eta(\mathcal{H}_1, \mathcal{H}_2) = 2^{n-2m}$.*

Proof. As $\eta(\mathcal{H}_1, \mathcal{H}_2) = 2^{n-\mathsf{rank}(H)}$, we need to construct parity-check matrices H_1 and H_2 for the codes $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathsf{rank}(H_1 \| H_2) = 2m$. We first show that there exists a $2m \times 2m$ binary matrix $A_m = A_1 \| A_2$, where A_1, A_2 are two $m \times 2m$ binary matrices whose columns are distinct and nonzero, such that $\mathsf{rank}(A_m) = 2m$. For $m = 3$, such a matrix is given by

$$A_3 \;=\; \left[\begin{array}{c} 1\,0\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0\,1 \\ 0\,0\,1\,1\,1\,0 \\ \hline 1\,0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,1\,1 \\ 0\,0\,1\,0\,1\,1 \end{array}\right].$$

For $m \ge 4$, we can construct A_m recursively as follows. Suppose that $A_{m-1} = A'_1 \| A'_2$, and take

$$A_m \;=\; \left[\begin{array}{c} A_1 \\ \hline A_2 \end{array}\right] \;=\; \left[\begin{array}{c|c|c} 1 & 0\cdots0 & 0 \\ \hline \mathbf{0} & A'_1 & x \\ \hline \mathbf{1} & A'_2 & \mathbf{0} \\ \hline 1 & 0\cdots0 & 1 \end{array}\right], \tag{3.9}$$

where x is any nonzero $(m-1)$-tuple that does not appear as a column of A'_1. It is easy to see from (3.9) that if A_{m-1} is a nonsingular matrix of rank $2(m-1)$, then A_m is a nonsingular matrix of rank $2m$. Now, since the columns of A_1 and A_2 are nonzero and distinct, these matrices can be extended, in an arbitrary manner, to parity-check matrices H_1 and H_2 of two Hamming codes of length $2^m - 1$. By construction, we have $\mathsf{rank}(H_1 \| H_2) = \mathsf{rank}(A_1 \| A_2) = 2m$. □

THEOREM 3.4. *For each $m \ge 3$, there exist two Hamming codes $\mathcal{H}_1, \mathcal{H}_2$ of length $n = 2^m - 1$, such that*

$$\eta(\mathcal{H}_1, \mathcal{H}_2) \;=\; 2^{n-r} \quad \textit{for} \quad r = m{+}1, m{+}2, \ldots, 2m.$$

Proof. The proof is by induction on m. The induction basis for $m = 3$ is established in (3.7). Now assume that, for each $r = m, m{+}1, \ldots, 2(m{-}1)$, there exist parity-check matrices H'_1 and H'_2 of two Hamming codes of length $2^{m-1} - 1$, such that $\mathsf{rank}(H'_1 \| H'_2) = r$. Take

$$H_1 \;=\; \left[\begin{array}{c|c|c} 0\cdots0 & 1 & 1\cdots1 \\ \hline H'_1 & \mathbf{0} & H'_1 \end{array}\right], \qquad H_2 \;=\; \left[\begin{array}{c|c|c} 0\cdots0 & 1 & 1\cdots1 \\ \hline H'_2 & \mathbf{0} & H'_2 \end{array}\right].$$

It is easy to see that H_1, H_2 are parity-check matrices of isomorphic Hamming codes of length $2^m - 1$, and that

$$\mathsf{rank}(H_1 \| H_2) \;=\; \mathsf{rank}(H_1' \| H_2') + 1 \;=\; r + 1.$$

Thus, all ranks in the range $r + 1 = m+1, m+2, \ldots, 2m-1$ are attainable. Finally, the rank of $2m$ is also attainable by Lemma 3.3, which completes the induction step. □

4. Embeddings and generalized Hamming weights. Let $\mathbb{C}_1$ be a code of length n_1, and let $\mathbb{C}_2$ be a code of length $n_2 \geq n_1$. We say that $\mathbb{C}_1$ is *embedded* in $\mathbb{C}_2$, in the first n_1 positions, if the code $\mathbb{C}_2$ punctured in the last $n_2 - n_1$ positions contains $\mathbb{C}_1$ as a subcode, and furthermore all the codewords of $\mathbb{C}_2$ that correspond to this subcode agree in the last $n_2 - n_1$ positions. This definition extends in the obvious way to any set of n_1 positions. Thus we say that $\mathbb{C}_1$ is *embedded* in $\mathbb{C}_2$ if it is embedded in some n_1 positions of $\mathbb{C}_2$. We note that our definition of embedding is a natural generalization of the concept of *shortening* (cf. [20, p. 29]) to nonlinear codes.

It is well known that any Hamming code of length $n = 2^m - 1$ contains a Hamming code of length $\nu = 2^{m-1}$–1 as a shortened subcode. Under which conditions is a similar assertion true for nonlinear perfect codes? Namely, when does a perfect code $\mathbb{C}$ of length $n = 2^m - 1$ contain a perfect code of length $\nu = 2^{m-1} - 1$ embedded in it? In what follows, we will prove that this happens if and only if $\mathbb{C}$ is not of full rank.

For a code $\mathbb{C} \subset \mathbb{F}_2^n$, we denote by $\mathbb{C}^\perp$ the subspace of $\mathbb{F}_2^n$ consisting of those vectors that are orthogonal to all the codewords of $\mathbb{C}$. It is obvious that $\dim \mathbb{C}^\perp + \dim \langle \mathbb{C} \rangle = n$, and therefore $\mathbb{C}$ is full rank if and only if $\mathbb{C}^\perp = \{\mathbf{0}\}$. The following observation, established in [9], will be key to our results in this section: for a perfect code $\mathbb{C}$ of length $n = 2^m - 1$, all the nonzero codewords in $\mathbb{C}^\perp$ have weight 2^{m-1}.

PROPOSITION 4.1. *If $\mathbb{C}$ is a perfect code of length $n = 2^m - 1$ and $\mathsf{rank}(\mathbb{C}) < n$, then there exists a perfect code of length $\nu = 2^{m-1} - 1$ embedded in $\mathbb{C}$.*

Proof. Let v be a codeword of $\mathbb{C}^\perp$ of weight 2^{m-1}. Without loss of generality, we can assume that $v = (\mathbf{1}|\mathbf{0})$ so that every codeword of $\mathbb{C}$ has even weight in the first 2^{m-1} positions. For $x \in \mathbb{E}_2^{\nu+1}$, define $\mathbb{C}_x = \{y : (x|\, y) \in \mathbb{C}\}$. Then either $\mathbb{C}_x = \varnothing$ or $\mathbb{C}_x$ is a code of length $\nu = 2^{m-1} - 1$ embedded in $\mathbb{C}$. We will show that, in fact, $\mathbb{C}_x$ is a perfect code for all x. Indeed, if $\mathbb{C}_x \neq \varnothing$, then $d(\mathbb{C}_x) \geq 3$ and therefore $|\mathbb{C}_x| \leq 2^{2^{m-1}-m}$. Hence,

$$2^{2^m - m - 1} \;=\; |\mathbb{C}| \;=\; \sum_x |\mathbb{C}_x| \;\leq\; 2^{2^{m-1}-m} \left|\mathbb{E}_2^{2^{m-1}}\right| \;=\; 2^{2^m - m - 1}. \tag{4.1}$$

Since (4.1) must hold with equality, for all $x \in \mathbb{E}_2^{2^{m-1}}$ we have $|\mathbb{C}_x| = 2^{2^{m-1}-m}$, and $\mathbb{C}_x$ is a perfect code of length ν embedded in $\mathbb{C}$. □

PROPOSITION 4.2. *If $\mathbb{C}$ is a perfect code of length $n = 2^m - 1$ and $\mathsf{rank}(\mathbb{C}) = n$, there is no perfect code of length $\nu = 2^{m-1} - 1$ embedded in $\mathbb{C}$.*

Proof. Assume to the contrary that $\mathbb{C}_1$ is a perfect code of length ν embedded in the last ν positions of $\mathbb{C}$. Then $\mathbb{C}$ contains $|\mathbb{C}_1| = 2^{\nu-(m-1)}$ codewords of the form $(a|c)$, where $a = (a_1, a_2, \ldots, a_{\nu+1})$ is a fixed $(\nu+1)$-tuple. Now let M be a $|\mathbb{C}| \times (\nu+1)$ matrix whose rows are the codewords of $\mathbb{C}$ truncated to the first $\nu + 1$ positions. For $x \in \mathbb{F}_2^\nu$, let $\omega_0(x)$, respectively, $\omega_1(x)$, denote the number of times $(x|0)$, respectively, $(x|1)$, appears as a row of M. Since $\mathbb{C}$ is an orthogonal array of strength ν (cf. [20, p. 139]), it is obvious that $\omega_0(x) + \omega_1(x) = |\mathbb{C}|/2^\nu = 2^{\nu-(m-1)}$ for

all x. Furthermore, it is shown in Proposition 4.2 of [9] that

$$\omega_0(x) = \begin{cases} \omega_0(\mathbf{0}) & \text{if } \mathrm{wt}(x) \equiv 0 \mod 2, \\ \omega_1(\mathbf{0}) & \text{if } \mathrm{wt}(x) \equiv 1 \mod 2. \end{cases} \tag{4.2}$$

Observe that since $\mathbb{C}$ contains the all-zero codeword $\mathbf{0}$, we have $\omega_0(\mathbf{0}) \neq 0$. Now consider $x = (a_1, a_2, \ldots, a_\nu)$; it is clear that either $\omega_0(x) = 2^{\nu-(m-1)}$ if $a_{\nu+1} = 0$, or $\omega_1(x) = 2^{\nu-(m-1)}$ if $a_{\nu+1} = 1$. In either case, this implies that $\omega_0(\mathbf{0}) = 2^{\nu-(m-1)}$ in view of (4.2) and the fact that $\omega_0(\mathbf{0}) \neq 0$. We can now count the number of even-weight rows of M, given by

$$\sum_{\substack{x \in \mathbb{F}_2^\nu \\ \mathrm{wt}(x) \equiv 0}} \omega_0(x) + \sum_{\substack{x \in \mathbb{F}_2^\nu \\ \mathrm{wt}(x) \equiv 1}} \omega_1(x) = 2^{\nu-1}\omega_0(\mathbf{0}) + 2^{\nu-1}\omega_0(\mathbf{0}) = 2^\nu \cdot 2^{\nu-(m-1)} = |\mathbb{C}|.$$

Thus *all* the codewords of $\mathbb{C}$ have even weight in the first $\nu+1$ positions, which implies that the vector $(\mathbf{1}|\mathbf{0})$ of weight $\nu+1$ is orthogonal to $\mathbb{C}$. Hence $\mathbb{C}^\perp \neq \{\mathbf{0}\}$, which contradicts the fact that $\mathbb{C}$ is of full rank. □

Propositions 4.1 and 4.2 show that those sets of positions, where a perfect code of length $\nu = 2^{m-1}-1$ is embedded in a perfect code $\mathbb{C}$ of length $n = 2^m - 1$, are in one-to-one correspondence with nonzero codewords of $\mathbb{C}^\perp$. In particular, we have the following corollary.

COROLLARY 4.3. *Let $\mathbb{C}$ be a perfect code of length $2^m - 1$. A perfect code of length $2^{m-1}-1$ is embedded in $\mathbb{C}$ if and only if $\mathbb{C}$ is not of full rank.*

The embedding problem considered above leads to another interesting question about perfect codes and generalized Hamming weights. The generalized Hamming weights were introduced by Wei [36] for linear codes and were studied by several authors; see [4, 10, 11, 13, 37], among others. We now review the definition of generalized Hamming weights in [36] and extend it to nonlinear codes.

The *support* of a code $\mathbb{C}$ of length n, denoted $\chi(\mathbb{C})$, is the set of positions i, such that there exist codewords $(c_1, c_2, \ldots, c_n)$, $(c'_1, c'_2, \ldots, c'_n) \in \mathbb{C}$ with $c_i \neq c'_i$. Notice that this definition of $\chi(\cdot)$, introduced in [17], applies to both linear and nonlinear codes; it coincides with the usual notion of support as the set of nonzero positions for linear codes. Now let $\mathbb{C}$ be a linear code of length n and dimension k. Then the ith generalized Hamming weight of $\mathbb{C}$ is defined [36] as

$$d_i(\mathbb{C}) \stackrel{\text{def}}{=} \min_D |\chi(D)| \quad \text{for} \quad i = 1, 2, \ldots, k, \tag{4.3}$$

where the minimum is taken over all linear subcodes $D \subset \mathbb{C}$ such that $\dim D = i$. The sequence $d_1(\mathbb{C}), d_2(\mathbb{C}), \ldots, d_k(\mathbb{C})$ is called the *generalized Hamming weight hierarchy* (GHW) of $\mathbb{C}$. This sequence plays an important role in many applications, ranging from the wire-tap channel [22] to trellis decoding [11]. For some of these applications, an equivalent sequence $\kappa_1(\mathbb{C}), \kappa_2(\mathbb{C}), \ldots, \kappa_n(\mathbb{C})$, called the *dimension-length profile* (DLP) of $\mathbb{C}$, is more convenient to deal with. This sequence, introduced in [34] and later studied in [11, 16, 17] and other works, is defined as follows:

$$\kappa_i(\mathbb{C}) \stackrel{\text{def}}{=} \max_D \dim D \quad \text{for} \quad i = 1, 2, \ldots, n, \tag{4.4}$$

where the maximum is taken over all linear subcodes $D \subset \mathbb{C}$ such that $|\chi(D)| = i$. The DLP and GHW are equivalent sequences, in the sense that either sequence can

be obtained from the other, as follows:

$$d_i(\mathbb{C}) = \min\{\, j \;:\; \kappa_j(\mathbb{C}) \geq i \,\} \quad \text{for} \quad i = 1, 2, \ldots, k, \tag{4.5}$$

$$\kappa_i(\mathbb{C}) = \max\{\, j \;:\; d_j(\mathbb{C}) \leq i \,\} \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{4.6}$$

A natural generalization of DLP and GHW to nonlinear codes is through the notion of *cardinality-length profile* (CLP), defined as follows. For any code $\mathbb{C} \subset \mathbb{F}_2^n$, we let

$$\kappa_i(\mathbb{C}) \stackrel{\text{def}}{=} \max_D \log_2 |D| \quad \text{for} \quad i = 1, 2, \ldots, n, \tag{4.7}$$

where the maximum is taken over all subcodes $D \subset \mathbb{C}$ such that $|\chi(D)| = i$. Thus $\kappa_i(\mathbb{C})$ is the log cardinality of the largest code of length i embedded in $\mathbb{C}$. The GHW of a nonlinear code $\mathbb{C}$ may be now defined[1] by (4.5), with $\lfloor \log_2 |\mathbb{C}| \rfloor$ replacing k.

The generalized Hamming weights of the Hamming codes were determined by Wei in [36]. Wei [36] showed that if $\mathcal{H}_m$ is a Hamming code of length $n = 2^m - 1$, then its GHW is given by $\{d_1(\mathbb{C}), d_2(\mathbb{C}), \ldots, d_k(\mathbb{C})\} = \{1, 2, \ldots, n\} \setminus \{1, 2, 2^2, \ldots, 2^{m-1}\}$. From this, it is easy to deduce that

$$\kappa_i(\mathcal{H}_m) \;=\; i - \lfloor \log_2 i \rfloor - 1 \quad \text{for} \quad i = 1, 2, \ldots, n.$$

In what follows, we show that certain nonlinear perfect codes, in particular the full-rank perfect codes, have a different cardinality-length profile. We also prove that the cardinality-length profile $\kappa_i(\mathbb{C})$ of any perfect code $\mathbb{C}$ of length $2^m - 1$ coincides with that of $\mathcal{H}_m$ for $i \geq 2^{m-1}$, and provide bounds on $\kappa_i(\mathbb{C})$ for other values of i. These bounds will enable us to conclude that the GHW of "most" perfect codes coincides with the GHW of the Hamming codes.

THEOREM 4.4. *Let $\mathbb{C}$ be a perfect code of length $n = 2^m - 1$. Then*

$$\kappa_i(\mathbb{C}) \;=\; i - m \quad \textit{for} \quad i = 2^{m-1}, 2^{m-1}+1, \ldots, 2^m - 1.$$

Proof. For these values of i, we have $n - i \leq 2^{m-1} - 1$. Since $\mathbb{C}$ is an orthogonal array of strength $2^{m-1} - 1$ (cf. [20, p. 139]), it follows that every set of $n-i$ positions of $\mathbb{C}$ contain each binary $(n-i)$-tuple exactly $|\mathbb{C}|/2^{n-i} = 2^{i-m}$ times. □

For $i = 2^{m-1}-1$, however, the CLP is *not* the same for all perfect codes. Specifically, if $\mathbb{C}$ is not of full rank, then $\kappa_{2^{m-1}-1}(\mathbb{C}) = 2^{m-1} - m$ by Proposition 4.1. If $\mathbb{C}$ is a full-rank perfect code, then $\kappa_{2^{m-1}-1}(\mathbb{C}) < 2^{m-1} - m$ by Proposition 4.2, since if a code D of length $2^{m-1} - 1$ and cardinality $2^{m-1} - m$ is embedded in $\mathbb{C}$, it must be a perfect code.

PROPOSITION 4.5. *If $\mathbb{C}$ is a full-rank perfect code of length $n = 2^m - 1$, then*

$$2^{m-1} - m - 1 \;<\; \kappa_{2^{m-1}-1}(\mathbb{C}) \;<\; 2^{m-1} - m.$$

Proof. The upper bound is Proposition 4.2. The lower bound may be proved as follows. For a vector $v \in \mathbb{F}_2^n$, let $\xi(v)$ be the number of codewords of $\mathbb{C}$ whose support is disjoint with the support of v. Further, let $\chi_v(\mathbb{C}) = \sum_{c \in \mathbb{C}} (-1)^{<v,c>}$ be the corresponding character of $\mathbb{C}$ (cf. [20, p. 134]). Now suppose that $\mathrm{wt}(v) = 2^{m-1}$. Then it follows from (4.2) that $\chi_v(\mathbb{C}) = 2^{2^{m-1}} \xi(v) - 2^{n-m}$. Since all perfect codes have the same weight distribution, the MacWilliams identities for nonlinear codes [6, 20] imply

$$\frac{1}{2^{n-m}} \sum_{\mathrm{wt}(v) = 2^{m-1}} \chi_v(\mathbb{C}) \;=\; \sum_{\mathrm{wt}(v) = 2^{m-1}} \left(\frac{\xi(v)}{2^{2^{m-1}-m-1}} - 1 \right) \;=\; 2^m - 1.$$

[1] The CLP was first introduced in [17]. For an alternative way to extend the definition of generalized Hamming weight hierarchy to nonlinear codes, see [4].

Hence, there exists at least one v of weight 2^{m-1} such that $\xi(v) > 2^{2^{m-1}-m-1}$. Now, it is obvious that $\kappa_{2^{m-1}-1}(\mathbb{C}) \geq \max_{\mathrm{wt}(v)=2^{m-1}} \log_2 \xi(v)$, and the lower bound follows. □

Establishing nontrivial bounds on the cardinality-length profile $\kappa_i(\mathbb{C})$ of full-rank perfect codes for $i \leq 2^{m-1}-1$ appears to be a difficult problem. On the other hand, we have the following.

THEOREM 4.6. *If a perfect code $\mathbb{C}$ of length $n = 2^m - 1$ has no full-rank perfect codes (of any length $\leq n$) embedded in it, then*

$$\begin{aligned} \kappa_i(\mathbb{C}) &\geq i - \lfloor \log_2 i \rfloor - 1 \quad \textit{for} \quad i = 1, 2, \ldots, 2^{m-1}-5, \\ \kappa_i(\mathbb{C}) &= i - \lfloor \log_2 i \rfloor - 1 \quad \textit{for} \quad i = 2^{m-1}-4, 2^{m-1}-3, \ldots, 2^m-1. \end{aligned}$$

Proof. Without loss of generality, assume that the dual code $\mathbb{C}^\perp$ contains the vector $(\mathbf{1}|\mathbf{0})$ of weight 2^{m-1}, and let $\mathbf{0}^i$ denote the all-zero i-tuple. It follows from the proof of Proposition 4.1 that $\mathbb{C}_1 = \{x : (\mathbf{0}^{2^{m-1}} | \, x) \in \mathbb{C}\}$ is a perfect code of length $2^{m-1}-1$. As such, it is an orthogonal array of strength $2^{m-2}-1$. Therefore, for each $i = 1, 2, \ldots, 2^{m-2}-1$, the set

$$D = \left\{ x : \left(\mathbf{0}^{2^{m-1}} | \, \mathbf{0}^i | \, x\right) \in \mathbb{C} \right\}$$

is a code of cardinality $|D| = |\mathbb{C}_1|/2^i = 2^{2^{m-1}-m-i}$ embedded in $\mathbb{C}$. Furthermore, the code $\mathbb{C}_1$ is not of full rank, by assumption. Hence, we can assume w.l.o.g. that $\mathbb{C}_1^\perp$ contains the vector $(\mathbf{1}|\mathbf{0})$ of weight 2^{m-2}. It follows, again by Proposition 4.1, that

$$\mathbb{C}_2 = \left\{ x : \left(\mathbf{0}^{2^{m-1}} | \, \mathbf{0}^{2^{m-2}} | \, x\right) \in \mathbb{C} \right\}$$

is a perfect code of length $2^{m-2}-1$ embedded in both $\mathbb{C}_1$ and $\mathbb{C}$. This code is again an orthogonal array and is not of full rank by assumption. Therefore, its dual code contains a vector of weight 2^{m-3}, and so on. Continuing in this manner until the length of $\mathbb{C}$ is exhausted, we obtain

$$\kappa_i(\mathbb{C}) \geq i - \lfloor \log_2 i \rfloor - 1 \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{4.8}$$

The equality in (4.8) for $i \geq 2^{m-1}-1$ follows from Theorem 4.4 and Proposition 4.1. The equality for $i = 2^{m-1}-4, 2^{m-1}-3, 2^{m-1}-2$ follows from the fact, established in [2], that triply shortened perfect codes are optimal. □

We conjecture that, in fact, equality always holds in (4.8). That is, the CLP of a perfect code $\mathbb{C}$ coincides with that of a Hamming code, provided there are no full-rank perfect codes embedded in $\mathbb{C}$. This is certainly true for perfect codes of length 15.

COROLLARY 4.7. *Let $\mathbb{C}$ be a perfect code of length 15. Then*

$$\kappa_i(\mathbb{C}) = i - \lfloor \log_2 i \rfloor - 1 \quad \textit{for} \quad i = 1, 2, \ldots, 15$$

if and only if $\mathbb{C}$ is not of full rank.

Proof. This follows from Proposition 4.5 and Theorem 4.6, along with the following observations: a perfect code of length 7 is necessarily a $(7, 4, 3)$ Hamming code; shortening the $(7, 4, 3)$ code any number of times produces optimal codes. □

Returning from the cardinality-length profiles to the generalized Hamming weights, Theorem 4.6 implies the following strong result.

THEOREM 4.8. *Let $\mathbb{C}$ be a perfect code of length $n = 2^m - 1$. Then*

$$\{d_1(\mathbb{C}), d_2(\mathbb{C}), \ldots, d_{n-m}(\mathbb{C})\} = \{1, 2, \ldots, n\} \setminus \{1, 2, 2^2, \ldots, 2^{m-1}\}, \tag{4.9}$$

provided there are no full-rank perfect codes embedded in $\mathbb{C}$.

Proof. Recall that the GHW of $\mathbb{C}$ is defined by (4.5). Thus, the theorem follows immediately from Theorem 4.6, along with the observation that $\kappa_i(\mathbb{C}) < i - \lfloor \log_2 i \rfloor$ for all i. The latter statement follows from the fact that an $(n, M, 3)$ code with $M \geq 2^{n-\lfloor \log_2 n \rfloor}$ does not exist by the sphere-packing bound [20, p. 19]. □

Finally, we observe that the generalized Hamming weight hierarchy of a full-rank perfect code is not given by (4.9), in view of Proposition 4.5.

5. Full-rank tilings and kernels of perfect codes. A *tiling* of $\mathbb{F}_2^n$ is a pair (V, A) of subsets of $\mathbb{F}_2^n$ such that every $x \in \mathbb{F}_2^n$ has a unique representation of the form $x = v + a$, with $v \in V$ and $a \in A$. Thus (V, A) is a tiling if and only if

$$V + A = \mathbb{F}_2^n \text{ and } (V + V) \cap (A + A) = \{\mathbf{0}\}.$$

Without loss of generality, we can always assume that $\mathbf{0} \in (V \cap A)$. A tiling (V, A) of $\mathbb{F}_2^n$ is *trivial* if one of the sets V, A is $\{\mathbf{0}\}$ and the other is $\mathbb{F}_2^n$. It is of *full rank* if $\langle V \rangle = \langle A \rangle$ or, equivalently, $\mathsf{rank}(V) = \mathsf{rank}(A) = n$. The study of [5] shows that any tiling of $\mathbb{F}_2^n$ can be uniquely decomposed into, or constructed from, smaller tilings that are either trivial or have full rank. Hence, the following question is of interest: for which values of n does $\mathbb{F}_2^n$ admit a full-rank tiling?

It is shown in [9, 5] that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 112$. In this section we show that, in fact, full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 14$. Two alternative constructions of such tilings are presented: an iterative "lifting" from a full-rank tiling of $\mathbb{F}_2^{14}$ exhibited in [5] and a direct reduction from a full-rank perfect code of length 1023. Since full-rank tilings of $\mathbb{F}_2^n$ do not exist for $n \leq 7$, as established in [5], these constructions leave only the six values $n = 8, 9, \ldots, 13$ unresolved. We will show that the existence of full-rank tilings for these values of n is closely related to the existence of full-rank perfect codes with kernels of high dimension. We start with the following iterative construction of tilings.

CONSTRUCTION C. *Let (V, A) be a tiling of $\mathbb{F}_2^n$ and let a^* be a nonzero element of A. Consider the sets*

$$V' = \{\, (v|0) : v \in V \,\} \cup \{\, (v|1) : v \in V \,\}, \tag{5.1}$$

$$A' = \{\, (a|0) : a \in A^* \,\} \cup \{\, (a^*|1) \,\}, \tag{5.2}$$

where $A^* = A \backslash \{a^*\}$. Then (V', A') is a tiling of $\mathbb{F}_2^{n+1}$.

Indeed, suppose that $x \in (V' + V') \cap (A' + A')$. Since $(V + V) \cap (A + A) = \{\mathbf{0}\}$, it follows that $x = (\mathbf{0}|0)$ or $x = (\mathbf{0}|1)$. But $(\mathbf{0}|1) \notin A' + A'$, which implies that $x = \mathbf{0}$. Furthermore, since $|V'| = 2|V|$ and $|A'| = |A|$, we have $|V'||A'| = 2^{n+1}$. Hence (V', A') is a tiling, as claimed.

PROPOSITION 5.1. *If (V, A) is a full-rank tiling of $\mathbb{F}_2^n$ and $\mathsf{rank}(A^*) = n$, then the tiling (V', A') obtained by* Construction C *is a full-rank tiling of $\mathbb{F}_2^{n+1}$.*

Proof. It is obvious from (5.1) that $\langle V \rangle = \mathbb{F}_2^n$ implies $\langle V' \rangle = \mathbb{F}_2^{n+1}$. Since $\mathsf{rank}(A^*) = n$, it follows that any vector of the form $(x|0)$, including $(a^*|0)$, belongs to $\langle A' \rangle$. Hence $(\mathbf{0}|1) = (a^*|0) + (a^*|1)$ also belongs to $\langle A' \rangle$, and therefore $\langle A' \rangle = \mathbb{F}_2^{n+1}$. □

A full-rank tiling (V, A) of $\mathbb{F}_2^{14}$ with $|V| = 2^{10}$ and $|A| = 2^4$ was constructed in [5]. We will call this the *seed tiling*. Starting with the seed tiling, and iteratively applying Construction C, establishes the following.

THEOREM 5.2. *For all $n \geq 14$, there exists a full-rank tiling of $\mathbb{F}_2^n$.*

Since we are interested here in the connections between tilings and perfect codes, we will now present an alternative proof of Theorem 5.2 which employs such connections. As a by-product, we will obtain certain bounds relating the rank of a perfect

code and the dimension of its kernel. The following theorem, established in [3, 5], is based on the matrix construction of covering of Blokhuis and Lam [3].

THEOREM 5.3. *Let (V, A) be a tiling of $\mathbb{F}_2^n$ and let $\nu = |V| - 1$. Further, let $H(V)$ be an $n \times \nu$ matrix having the nonzero elements of V as its columns. Define*

$$\mathbb{C} \;=\; \{\, x \in \mathbb{F}_2^{\nu} \;:\; H(V)x^t \in A \,\}.$$

Then $\mathbb{C}$ is a perfect code of length ν.

We shall say that $\mathbb{C}$ is the perfect code *associated* with the tiling (V, A). The following relation between the ranks of V, A, and $\mathbb{C}$ was established in [5].

PROPOSITION 5.4. *If $\mathbb{C}$ is the perfect code of length ν associated with a tiling (V, A), then*

$$\mathsf{rank}(\mathbb{C}) \;=\; \nu - \mathsf{rank}(V) + \mathsf{rank}(A_{\langle V\rangle}),$$

where $A_{\langle V\rangle} = A \cap \langle V\rangle$. In particular, if $\langle V\rangle = \mathbb{F}_2^n$, then

$$\mathsf{rank}(\mathbb{C}) \;=\; \nu - n + \mathsf{rank}(A).$$

It follows from Proposition 5.4 that if $\langle V\rangle = \langle A\rangle = \mathbb{F}_2^n$, then $\mathsf{rank}(\mathbb{C}) = \nu$. Thus if (V, A) is a full-rank tiling, then the associated perfect code $\mathbb{C}$ is also of full rank.

Given a code $\mathbb{C} \subset \mathbb{F}_2^n$, the *kernel* of $\mathbb{C}$ is the set of all $x \in \mathbb{F}_2^n$ that leave $\mathbb{C}$ invariant under translation. Assuming that $\mathbf{0} \in \mathbb{C}$, the kernel of $\mathbb{C}$ can be defined (cf. [1, 26]) as follows:

$$\mathsf{ker}\,\mathbb{C} \;\stackrel{\text{def}}{=}\; \{\, x \in \mathbb{C} \;:\; x + \mathbb{C} = \mathbb{C} \,\}.$$

It is easy to see that $\mathsf{ker}\,\mathbb{C}$ is a linear subcode of $\mathbb{C}$, and $\mathsf{ker}\,\mathbb{C} = \mathbb{C}$ if and only if $\mathbb{C}$ itself is linear. The kernel of $\mathbb{C}$ is sometimes called the set of stabilizers of $\mathbb{C}$ (cf. [14]) or the set of periodic points of $\mathbb{C}$ (cf. [5]).

Now let $\mathbb{C}$ be the perfect code associated with a tiling (V, A). Then it is easy to see that

$$\mathsf{ker}\,\mathbb{C} \;=\; \{\, x \in \mathbb{C} \;:\; H(V)x^t \in \mathsf{ker}\,A \,\}.$$

Along with Proposition 5.4, this immediately implies the following.

PROPOSITION 5.5. *If $\mathbb{C}$ is the perfect code of length ν associated with a tiling (V, A), then*

$$\dim(\mathsf{ker}\,\mathbb{C}) \;=\; \nu - \mathsf{rank}(V) + \dim(\mathsf{ker}\,A_{\langle V\rangle}),$$

where $A_{\langle V\rangle} = A \cap \langle V\rangle$. In particular, if $\langle V\rangle = \mathbb{F}_2^n$, then

$$\dim(\mathsf{ker}\,\mathbb{C}) \;=\; \nu - n + \dim(\mathsf{ker}\,A).$$

Kernels play an important role in the construction of tilings introduced in [5]. We now briefly describe this construction.

Let A_0 be a subspace of $\mathbb{F}_2^n$ of dimension k. For any $V \subset \mathbb{F}_2^n$, we define V/A_0 as follows. Fix a basis $a_1, a_2, \ldots, a_k$ for A_0 and complete this to a basis $a_1, a_2, \ldots, a_k$, $b_1, b_2, \ldots, b_{n-k}$ for $\mathbb{F}_2^n$. Then each vector $v = \sum_{i=1}^{k} \alpha_i a_i + \sum_{i=1}^{n-k} \beta_i b_i$ in V is mapped onto the vector $v' = \sum_{i=1}^{n-k} \beta_i b_i$ in V/A_0. Thus V/A_0 is just the projection of V onto $\mathbb{F}_2^n/A_0$. Note that $\mathbb{F}_2^n/A_0$ may be regarded as $\mathbb{F}_2^{n-k}$ under an appropriate change of

basis (cf. [5]), namely, under the linear transformation that takes $b_1, b_2, \ldots, b_{n-k}$ into unit vectors. Thus we will identify $\mathbb{F}_2^n/A_0$ with $\mathbb{F}_2^{n-k}$ and think of V/A_0 as a subset of $\mathbb{F}_2^{n-k}$.

CONSTRUCTION D. *Let (V, A) be a tiling of $\mathbb{F}_2^n$. Further, let A_0 be a k-dimensional subspace of $\ker A$. Then $(V/A_0, A/A_0)$ is a tiling of $\mathbb{F}_2^{n-k}$.*

It is shown in [5] that if (V, A) is a full-rank tiling, then so is $(V/A_0, A/A_0)$. This implies the following.

PROPOSITION 5.6. *If there exists a full-rank tiling (V, A) of $\mathbb{F}_2^n$ with $\dim(\ker A) = r$, then there exist full-rank tilings of $\mathbb{F}_2^{n-k}$ for all $k = 1, 2, \ldots, r$.*

Propositions 5.4–5.6 and Theorem 5.3 provide an alternative proof for Theorem 5.2 as follows. Consider again the seed full-rank tiling (V, A) of $\mathbb{F}_2^{14}$ exhibited in [5]. Recall that for this tiling $|V| = 2^{10}$, $|A| = 2^4$, and $\ker V = \ker A = \{\mathbf{0}\}$. By Theorem 5.3 and Proposition 5.4, the associated perfect code $\mathbb{C}$ is a full-rank code of length $2^{10} - 1 = 1023$. By Proposition 5.5, we have

$$\dim(\ker \mathbb{C}) \;=\; 1023 - \mathsf{rank}(V) + \dim(\ker A) \;=\; 1023 - 14 \;=\; 1009.$$

Now let $\mathcal{V}_n$ denote the Hamming sphere of radius 1 in $\mathbb{F}_2^n$. Then $(\mathcal{V}_{1023}, \mathbb{C})$ is obviously a full-rank tiling of $\mathbb{F}_2^{1023}$. Applying to this tiling Construction D and Proposition 5.6, we obtain full-rank tiling of $\mathbb{F}_2^n$ for all $n = 14, 15, \ldots, 1022$. On the other hand, it was already shown in [5] that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 112$.

Kernels of perfect binary codes were studied by Phelps and LeVan in [26]. It is shown in [26] that given $m \geq 4$ and $n = 2^m - 1$, there exists a nonlinear perfect code $\mathbb{C}$ of length n with kernel of dimension k, if and only if $k = 1, 2, \ldots, n-m-2$. However, if we also impose constraints on the rank of $\mathbb{C}$, for example require that $\mathbb{C}$ is of full rank, much less is known about the possible dimensions of its kernel. Propositions 5.4–5.6 shed some light on this problem. For example, starting with the full-rank tiling $(\mathcal{V}_{1023}, \mathbb{C})$ of $\mathbb{F}_2^{1023}$ discussed in the foregoing paragraph, and applying Construction D, yields associated full-rank perfect codes of length $n = 2^m - 1$ with kernels of dimension $\geq n - m - 10$ for $m = 4, 5, \ldots, 1022$. Furthermore, the code $\mathbb{C}$ itself, associated with the seed tiling, has kernel of dimension $n - m - 4$ for $m = 10$. The following theorem shows that this is the highest possible kernel dimension for a full-rank perfect code.

THEOREM 5.7. *If $\mathbb{C}$ is a full-rank perfect code length $n = 2^m - 1$, then*

$$\dim(\ker \mathbb{C}) \;\leq\; n - m - 4. \tag{5.3}$$

Furthermore, this bound is tight for $m = 10$ and $m = 11$.

Proof. Let $A_0 = \ker \mathbb{C}$, and assume to the contrary that $\dim A_0 \geq n - m - 3$. Obviously $(\mathcal{V}_n, \mathbb{C})$ is a full-rank tiling. Applying to this tiling Construction D, we obtain another full-rank tiling $(V, A) = (\mathbb{C}/A_0, \mathcal{V}_n/A_0)$ with

$$|V| \;=\; |\mathbb{C}/A_0| \;=\; \frac{|\mathbb{C}|}{|A_0|} \;\leq\; \frac{2^{n-m}}{2^{n-m-3}} \;=\; 8.$$

By Theorem 5.3 and Proposition 5.4, the perfect code associated with (V, A) must be a full-rank perfect code of length $|V| - 1 \leq 7$. But such a code obviously does not exist. The tightness of (5.3) for $m = 11$ follows by considering the perfect code associated with the tiling $(\mathbb{C}, \mathcal{V}_{15})$, where $\mathbb{C}$ is a full-rank perfect code of length 15. □

More generally, one could ask: What is the largest possible dimension $\alpha(m)$ of the kernel of a full-rank perfect code of length $n = 2^m - 1$? The following theorem provides a complete answer to this question for all $m \geq 10$.

THEOREM 5.8. *Let δ be the unique integer such that $2^{\delta-1} - (\delta-1) \leq m < 2^\delta - \delta$. Then*

$$\alpha(m) \;=\; 2^m - m - \delta - 1 \quad \textit{for} \quad m = 10, 11, \ldots. \tag{5.4}$$

Proof. We first show that $\alpha(m) \leq 2^m - m - \delta - 1 = n - (m+\delta)$, where $n = 2^m - 1$. Assume to the contrary that there exists a full-rank perfect code $\mathbb{C}$ of length n such that $\dim(\ker\mathbb{C}) = 2^m - m - \delta$. Observe that $\mathbb{C}$ is the union of $|\mathbb{C}|/|\ker\mathbb{C}|$ cosets of $\ker\mathbb{C}$. Hence, the total number of linearly independent vectors in $\mathbb{C}$ is at most

$$\dim(\ker\mathbb{C}) \;+\; \left(\frac{|\mathbb{C}|}{|\ker\mathbb{C}|} - 1\right) \;\geq\; n, \tag{5.5}$$

where the inequality follows from the assumption that $\mathbb{C}$ is of full rank. Substituting $\dim(\ker\mathbb{C}) = 2^m - m - \delta$ and $|\mathbb{C}| = 2^{n-m}$ into (5.5), we obtain $m \leq 2^{\delta-1} - \delta$, which contradicts the definition of δ. In conjunction with the result of Theorem 5.7, this proves (5.4) for $m = 10$ and $m = 11$.

Next, we show how to construct a full-rank perfect code $\mathbb{C}_{12}$ of length $n = 2^{12} - 1$, such that $\dim(\ker\mathbb{C}) = n - 17 = (2^m - 1) - m - \delta$, for $m = 12$. Start with the full-rank tiling $(V, A) = (\mathcal{V}_{15}, \mathbb{C})$ of $\mathbb{F}_2^{15}$, where $\mathbb{C}$ is a full-rank perfect code of length 15. Then apply Construction C to obtain a full-rank tiling (V', A') of $\mathbb{F}_2^{16}$ with $|V'| = 2^5$ and $|A'| = 2^{11}$. Now, apply Construction C again, with the roles of V' and A' interchanged. This produces a full-rank tiling (V_{12}, A_{12}) of $\mathbb{F}_2^{17}$ with $|V_{12}| = 2^{12}$ and $|A_{12}| = 2^5$. The full-rank perfect code $\mathbb{C}_{12}$ associated with this tiling has length $n = |V_{12}| - 1 = 2^{12} - 1$. Furthermore, by Proposition 5.5 we have

$$\dim(\ker\mathbb{C}_{12}) \;\geq\; n - \mathsf{rank}(V_{12}) \;=\; n - 17.$$

In view of the upper bound on $\alpha(m)$ that we have already proved, the above expression holds with equality. Thus, we have established (5.4) for $m = 12$. Now, iteratively applying Construction C to (V_{12}, A_{12}), we obtain full-rank tilings (V_m, A_m) of $\mathbb{F}_2^{m+5}$ with associated full-rank perfect codes of length $n = 2^m - 1$ and kernel of dimension $n - (m+5)$. Since in all of these tilings $|A_m| = |A_{12}| = 2^5$, we can keep iterating Construction C in this way as long as $m + 5 \leq 2^5 - 1$ or, equivalently, $m < 2^5 - 5 = 27$. This proves (5.4) for all $m = 12, 13, \ldots, 26$. For $m = 27, 28, \ldots, 57$, we start with the full-rank tiling $(\mathcal{V}_{31}, \mathbb{C})$, where $\mathbb{C}$ is a full-rank perfect code of length 31, and proceed as before. Continuing in this manner establishes (5.4) for all $m \geq 10$. □

Note that Theorem 5.7 is not a special case of Theorem 5.8, since it holds also for $m < 10$. For example, for $m = 4$ it follows from Theorem 5.7 that the possible dimensions of the kernel of a full-rank perfect code of length 15 are $1, 2, \ldots, 7$. The problem of determining which of these kernel dimensions are attainable is closely related to the problem of existence of full-rank tilings of $\mathbb{F}_2^n$ for $n = 8, 9, \ldots, 13$. Indeed, a full-rank perfect code of length 15 and kernel of dimension k implies by Proposition 5.6 the existence of a full-rank tilings of $\mathbb{F}_2^n$ for all $n \geq 15 - k$. Furthermore, we have the following result.

PROPOSITION 5.9. *A full-rank perfect code of length* 15 *with kernel of dimension* 7 *exists if and only if a full-rank tiling of $\mathbb{F}_2^8$ exists.*

Proof. Suppose that (V, A) is a full-rank tiling of $\mathbb{F}_2^8$. Then clearly $|V| = |A| = 16$. Hence, by Propositions 5.4 and 5.6, the associated perfect code has length 15, is of full rank, and has kernel of dimension $15 - \mathsf{rank}(V) = 7$. □

The linear code $\mathcal{A}$ defined in (2.1) plays a prominent role in the construction of full-rank perfect codes in [9] and has dimension 7 for $n = 15$. A generator matrix for $\mathcal{A}$ is given by

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix}.$$

However, we now show that $\mathcal{A}$ cannot be the kernel of a full-rank perfect code $\mathbb{C}$ of length 15. Indeed, assume to the contrary that this is so. Then $(\mathcal{V}_{15}/\mathcal{A}, \mathbb{C}/\mathcal{A})$ is a full-rank tiling of $\mathbb{F}_2^8$ by Proposition 5.6. Since both $\mathcal{V}_{15}$ and $\mathcal{A}$ are known, we can compute $\mathcal{V}_{15}/\mathcal{A}$ explicitly to obtain

$$\mathcal{V}_{15}/\mathcal{A} = \left\{ \begin{array}{llll} 0\,0\,0\,0\,0\,0\,0\,0, & 0\,0\,0\,1\,0\,0\,0\,1, & 1\,0\,0\,0\,0\,0\,0\,0, & 0\,0\,0\,0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0\,0\,0\,1, & 0\,0\,0\,0\,1\,0\,0\,1, & 0\,1\,0\,0\,0\,0\,0\,0, & 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,1, & 0\,0\,0\,0\,0\,1\,0\,1, & 0\,0\,1\,0\,0\,0\,0\,0, & 0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,0\,0\,1, & 0\,0\,0\,0\,0\,0\,1\,1, & 0\,0\,0\,1\,0\,0\,0\,0, & 0\,0\,0\,0\,0\,0\,0\,1 \end{array} \right\}.$$

We now observe that $\mathsf{ker}(\mathcal{V}_{15}/\mathcal{A}) = \{(00000000), (00000001)\}$ has dimension 1. In view of Proposition 5.6, this implies the existence of a full-rank tiling of $\mathbb{F}_2^7$. But such a tiling does not exist, as shown in [5].

Remark. LeVan and Phelps [25] have recently found full-rank perfect codes of length 15 with kernels of dimension 2, 3, 4, and 5. This, along with the results of this section, implies that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 10$.

6. Open problems. We have considered herein three topics concerning perfect codes and tilings: the intersection number problem; embeddings and generalized Hamming weights; and full-rank tilings and kernels of full-rank perfect codes. Solutions to some of these problems are provided in the foregoing three sections. Nevertheless, it is fair to say that we know much less than we would like to, and many problems concerning perfect codes remain open. We conclude this paper with a list of ten open problems on perfect binary codes which, at least in our opinion, seem to be the most interesting.

Intersection numbers. For a given m, what are the possible intersection numbers of distinct perfect codes of length $n = 2^m - 1$? For more details on this problem, see section 3.

GHW and CLP. Give a complete characterization of the generalized Hamming weights and/or the cardinality length profiles for perfect codes. Compare the generalized Hamming weight hierarchies for full-rank and not full-rank perfect codes, derived from different constructions. For more details on this problem, see section 4.

Full-rank tilings. Construct full-rank tilings of $\mathbb{F}_2^n$ for $n = 8$ and $n = 9$, or prove that such tilings do not exist. This problem appears to be quite challenging despite the small size of the sets involved. For more details on this, see section 5 and [5].

Rank and kernel. Given a perfect code $\mathbb{C}$ of length $n = 2^m - 1$, its rank r is in the range $n-m, \ldots, n$, while the dimension k of its kernel is in the range $1, \ldots, n-m-2$ or $n-m$. Furthermore, as shown in [9] and [26], each value of r or k in the corresponding range is attainable. We ask: which pairs (r, k) are attainable as the rank and kernel dimension of a perfect code of length $2^m - 1$? For bounds, and more details, see section 5.

Systematicity. A binary code $\mathbb{C}$ with 2^k codewords is called systematic if there exists a set of k positions in which every binary k-tuple appears (exactly once) among

the codewords of $\mathbb{C}$. Thus $\mathbb{C}$ is systematic if there exist some k positions that can be used as information positions for the code. All *known* perfect codes are systematic, and a longstanding conjecture says that *all* perfect codes are systematic. This conjecture is related to certain results on systems of t-resilient functions [18]. The systematicity problem was posed as open in an earlier version of this paper. It has been recently solved by Solov'eva and Avgustinovich [31] who showed that the systematicity conjecture is false: they proved that for each $n = 2^m - 1$ with $m \geq 6$, there exists a nonsystematic perfect code. Phelps and LeVan [27] have extended this result to all $m \geq 4$.

Enumeration. Classification of inequivalent perfect codes was first posed as a research problem in [20, p. 180]. However, it soon became apparent [23] that an exact classification is intractable. On the other hand, asymptotic bounds on the *number* of inequivalent perfect codes of length $n = 2^m - 1$ are known. A lower bound of $2^{2^{0.5n}}$ for sufficiently large n is given in [9, 24], while an upper bound of $2^{2^{n-m}}$ can be easily derived. The gap is very large and any improvement on these bounds would be an important result.

Optimality of shortening. It is established in [2] that triply shortened perfect codes of length $2^m - 1$ are optimal. That is, the number of codewords in these codes achieves the value of $A(n, 3)$ for $n = 2^m - 2, 2^m - 3, 2^m - 4$. Referring to the table of best known codes [19] suggests that shortening a perfect code of length $2^m - 1$ up to $2^{m-2} - 1$ times is likely to produce optimal codes for $m \leq 9$. However, the result of Kabatiansky and Panchenko [15] shows that this is not true in general for large m. Thus we ask: What is the largest integer s_m such that shortening a perfect code of length $2^m - 1$ up to s_m times produces optimal codes?

Uniqueness of shortening. Shortening a perfect code of length $2^m - 1$ once, that is, taking all the codewords that coincide in a fixed coordinate, produces a code of length $n = 2^m - 2$, with 2^{n-m} codewords and minimum Hamming distance 3. Now, we ask the reverse question: Given a code $\mathbb{C}$ of length $n = 2^m - 2$ with $|\mathbb{C}| = 2^{n-m}$ and minimum Hamming distance 3, is it always possible to extend $\mathbb{C}$ to a perfect code of length $2^m - 1$? The same question can be asked for shortening by more than one coordinate.

Uniqueness of STS. It is known that the codewords of weight 3 in a perfect code of length $n = 2^m - 1$ form a Steiner triple system (STS) of order n. Again we ask the reverse question: Can any Steiner triple system of order $n = 2^m - 1$ be extended to a perfect code of length n? A solution even for the fist case $n = 15$, would be very interesting. This problem was considered by Phelps in [23].

Space partitions. Finally, we suggest the following question. Given a perfect code $\mathbb{C}$ of length $n = 2^m - 1$, we know that there always exist $n+1$ translates of $\mathbb{C}$, say $\mathbb{C}_0, \mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_n$ with $\mathbb{C}_0 = \mathbb{C}$, that form a partition of $\mathbb{F}_2^n$. Under which conditions is there another, different, partition of $\mathbb{F}_2^n$ into perfect codes $D_0, D_1, D_2, \ldots, D_n$ with $D_0 = \mathbb{C}$? Can such partitions be classified for a given perfect code $\mathbb{C}$?

Acknowledgments. We wish to thank Simon Litsyn for the preprint of [19]. We are grateful to Noga Alon, Kevin Phelps, and Faina Solov'eva for stimulating discussions.

REFERENCES

[1] H. Bauer, B. Ganter, and F. Hergert, *Algebraic techniques for nonlinear codes*, Combinatorica, 3 (1983), pp. 21–33.

[2] M.R. Best and A.E. Brouwer, *The triply shortened binary Hamming code is optimal*, Discrete Math., 17 (1977), pp. 235–245.
[3] A. Blokhuis and C.W.H. Lam, *More coverings by rook domains*, J. Combin. Theory Ser. A, 36 (1984), pp. 240–244.
[4] G.D. Cohen, S. Litsyn, and G. Zémor, *Upper bounds on generalized distances*, IEEE Trans. Inform. Theory, 40 (1994), pp. 2090–2092.
[5] G.D. Cohen, S. Litsyn, A. Vardy, and G. Zémor, *Tilings of binary spaces*, SIAM J. Discrete Math., 9 (1996), pp. 393–412.
[6] Ph. Delsarte, *Four fundamental parameters of a code and their combinatorial significance*, Inform. Control, 23 (1973), pp. 407–438.
[7] Ph. Delsarte and J.-M. Goethals, *Unrestricted codes with the Golay parameters are unique*, Discrete Math., 12 (1975), pp. 211–224.
[8] T. Etzion, *Nonequivalent q-ary perfect codes*, SIAM J. Discrete Math., 9 (1996), pp. 413–423.
[9] T. Etzion and A. Vardy, *Perfect codes: Constructions, properties and enumeration*, IEEE Trans. Inform. Theory, 40 (1994), pp. 754–763.
[10] G.-L. Feng, K.K. Tzeng, and V.K. Wei, *On the generalized Hamming weights of several classes of cyclic codes*, IEEE Trans. Inform. Theory, 38 (1992), pp. 133–1140.
[11] G.D. Forney, Jr., *Dimension/length profiles and trellis complexity of linear block codes*, IEEE Trans. Inform. Theory, 40 (1994), pp. 1741–1752.
[12] O. Heden, *A binary perfect code of length* 15 *and codimension* 0, Des. Codes Cryptogr., 4 (1994), pp. 213–220.
[13] T. Helleseth, T. Kløve, and Ø. Ytrehus, *Generalized Hamming weights of linear codes*, IEEE Trans. Inform. Theory, 38 (1992), pp. 1412–1418.
[14] T.W. Hungerford, *Algebra*, Holt, Rinehart and Winston, New York, 1974.
[15] G. Kabatiansky and V. Panchenko, *Packings and coverings of the Hamming space by spheres of radius one*, Probl. Peredachi Inform., 24 (1988), pp. 3–16.
[16] A.B. Kiely, S. Dolinar, R.J. McEliece, L. Ekroot, and W. Lin, *Trellis decoding complexity of linear block codes*, IEEE Trans. Inform. Theory, 42 (1996), pp. 1687–1697.
[17] A. Lafourcade and A. Vardy, *Lower bounds on trellis complexity of block codes*, IEEE Trans. Inform. Theory, 41 (1995), pp. 1938–1954.
[18] V.I. Levenshtein, *private communication*, 1994.
[19] S. Litsyn, *An updated table of best known binary codes*, preprint, December 1995.
[20] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland, Amsterdam, 1977.
[21] M. Mollard, *A generalized parity function and its use in the construction of perfect codes*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 113–115.
[22] L.H. Ozarow and A.D. Wyner, *Wire-tap-channel II*, Bell Labs Tech. J., 63 (1984), pp. 2135–2157.
[23] K.T. Phelps, *A combinatorial construction of perfect codes*, SIAM J. Alg. Disc. Meth., 4 (1983), pp. 398–403.
[24] K.T. Phelps, *A general product construction for error-correcting codes*, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 224–228.
[25] K.T. Phelps, *private communication*, Auburn University, Auburn, AL, 1996.
[26] K.T. Phelps and M. LeVan, *Kernels of nonlinear Hamming codes*, Des. Codes Cryptogr., 6 (1995), pp. 247–257.
[27] K.T. Phelps and M. LeVan, *Non-systematic perfect codes*, preprint, 1996.
[28] V. Pless, *On the uniqueness of the Golay codes*, J. Combin. Theory, 5 (1968), pp. 215–228.
[29] S.L. Snover, *The Uniqueness of the Nordstrom-Robinson and the Golay Binary Codes*, Ph.D. Thesis, Dept. of Mathematics, Michigan State Univ., East Lansing, MI, 1973.
[30] F.I. Solov'eva, *On binary nongroup codes*, Metody Diskret. Anal., 37 (1981), pp. 65–76 (in Russian).
[31] F.I. Solov'eva and S.V. Avgustinovich, *Existence of nonsystematic perfect binary codes*, in Proc. Fifth Intl. Workshop on Algebraic and Combinatorial Coding Theory, Cosopol, Bulgaria, June 1996, pp. 15–19.
[32] A. Tietäväinen, *On the nonexistence of perfect codes over finite fields*, SIAM J. Appl. Math., 24 (1973), pp. 88–96.
[33] J.H. vanLint, *Nonexistence theorems for perfect error-correcting-codes*, in Computers in Algebra and Number Theory, vol. IV, SIAM–AMS Proceedings, SIAM, Philadelphia, 1971.
[34] A. Vardy and Y. Be'ery, *Maximum-likelihood soft decision decoding of BCH codes*, IEEE Trans. Inform. Theory, 40 (1994), pp. 546–554.
[35] J.L. Vasil'ev, *On nongroup close-packed codes*, Probl. Kibernet., 8 (1962), pp. 375–378 (in Russian).

[36] V.K. Wei, *Generalized Hamming weights for linear codes*, IEEE Trans. Inform. Theory, 37 (1991), pp. 1412–1418.
[37] V.K. Wei and K. Yang, *On the generalized Hamming weights of product codes*, IEEE Trans. Inform. Theory, 39 (1993), pp. 1709–1713.
[38] V.A. Zinov'ev and V.K. Leont'ev, *The nonexistence of perfect codes over Galois fields*, Probl. Control and Inform. Theory, 2 (1973), pp. 123–132 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 224–240, May 1998

004

SORTING BY TRANSPOSITIONS*

VINEET BAFNA† AND PAVEL A. PEVZNER‡

Abstract. Sequence comparison in computational molecular biology is a powerful tool for deriving evolutionary and functional relationships between genes. However, classical alignment algorithms handle only local mutations (i.e., insertions, deletions, and substitutions of nucleotides) and ignore global rearrangements (i.e., inversions and transpositions of long fragments). As a result, the applications of sequence alignment to analyze highly rearranged genomes (i.e., herpes viruses or plant mitochondrial DNA) are rather limited. The paper addresses the problem of *genome* comparison versus classical *gene* comparison and presents algorithms to analyze rearrangements in genomes evolving by *transpositions*. In the simplest form the problem corresponds to *sorting by transpositions*, i.e., sorting of an array using transpositions of arbitrary fragments. We derive lower bounds on *transposition distance* between permutations and present approximation algorithms for sorting by transpositions. The algorithms also imply a nontrivial upper bound on the *transposition diameter* of the symmetric group. Finally, we formulate two *biological* problems in genome rearrangements and describe the first *algorithmic* steps toward their solution.

Key words. computational molecular biology, genome rearrangements, transpositions, the symmetric group, approximation algorithm

AMS subject classifications. 15A15, 15A09, 15A23

PII. S089548019528280X

1. Introduction.

Studies of molecular evolution of herpes viruses raised many more questions than they answered. Genomes of herpes viruses evolve so rapidly that the extremes of present-day phenotypes may appear quite unrelated. As a result, the similarity between many genes in herpes viruses is so low that it is frequently indistinguishable from the background noise (Karlin, Mocarski, and Schachtel [16]). In particular, there is little or no cross-hybridization between DNAs of Epstein–Barr virus EBV and Herpes simplex virus HSV-1 and until recently there was no unambiguous evidence that these herpes viruses actually had a common evolutionary origin (McGeoch [20]). As a result the classical methods of *sequence comparison* are not very useful for such highly diverged genomes and the ventures into the quagmire of molecular phylogeny of herpes viruses may lead to contradictions, since different genes give rise to different evolutionary trees (Griffin and Boursnell [11]). However, recently a new approach to analyze highly diverged genomes was proposed, based on comparison of *gene orders* versus traditional comparison of *DNA sequences* (Sankoff et al. [24]). Since it is often found that the order of genes is much more conserved than the DNA sequence (Franklin [9]) this approach seems to be a method of choice for many "hard-to-analyze" genomes.

*Received by the editors March 10, 1995; accepted for publication (in revised form) June 2, 1997. A preliminary version of this paper appeared in *Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, SIAM, Philadelphia, PA, 1995, pp. 614–623.

http://www.siam.org/journals/sidma/11-2/28280.html

†Bioinformatics, SmithKline Beecham, 709 Swedeland Road, King of Prussia, PA 19406 (bafnav1@mh.us.sbphrd.com). Most of this author's research was carried out while he was at the Pennsylvania State University, University Park, PA and at DIMACS, Piscataway, NJ.

‡Departments of Mathematics and Computer Science, University of Southern California, DRB 155, Los Angeles, CA 90089-1113 (ppevzner@hto-a.usc.edu). The research of this author was supported by NIH grant 1R01 HG00987-01, by an NSF Young Investigator Award, and by NSF grant CCR-9308567.

Analysis of genomes of EBV and HSV-1 reveals that evolution of these herpes viruses involved a number of *inversions* and *transpositions* of large fragments; in particular, an analogue of the gene UL52-BSLF1 (required for DNA replication) in common herpes virus precursor "jumped" from one location in the genome to another (biologists call this event a transposition). The analysis of such rearrangements at the *genome* level might be more conclusive than the analysis at the *gene* level traditionally used in molecular evolution. However, there are almost no computer science results allowing a biologist to analyze genome rearrangements.

Genomes evolve by inversions and transpositions as well as by more simple operations of deletion, insertion, and duplication of fragments. Inversions seem to be a very common rearrangement; in fact, some genomes (for example, many plant mitochondrial DNA) are believed to evolve almost solely by inversions (Palmer and Herbon [23]). A combinatorial problem of *sorting by reversals* (corresponding to genome rearrangements by inversions) has been studied intensively in recent years, and currently there are two software programs which prove to be useful for analyzing rearrangements in animal (Sankoff et al. [24]) and plant (Bafna and Pevzner [3]) organelle DNA. In 1992 Kececiouglu and Sankoff suggested the first performance guarantee algorithm for sorting by reversal (see [17]). Later Bafna and Pevzner [2] devised a 1.75 performance guarantee algorithm for sorting by reversals and proved Gollan's conjecture on the reversal diameter of the symmetric group. See also Kececioglu and Ravi [18] and Hannenhalli and Pevzner [13] for recent progress on genome rearrangements. An interesting problem related to sorting by reversals is the problem of *sorting by prefix reversals*, also known as the *pancake flipping problem* (Gates and Papadimitriou [10]). Improved bounds for sorting by prefix reversals have been obtained recently (see Cohen and Blum [4]; Heydari and Sudborough [14]).

In a study of herpes viruses, Hannenhalli et al. [12] faced the problem of analyzing an entire spectrum of genome rearrangements—in particular, transpositions. As a first approximation, transpositions in genome rearrangements can be modeled in a straightforward but limited manner by *sorting by transpositions*, described below.

We assume that the order of genes in a genome is represented by a permutation $\pi = \pi_1\pi_2, \ldots, \pi_n$. Extend the permutation to include $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. For a permutation π, a *transposition* $\rho(i, j, k)$ (defined for all $1 \le i < j \le n + 1$ and all $1 \le k \le n + 1$ such that $k \notin [i, j]$) "inserts" an interval $[i, j - 1]$ of π between π_{k-1} and π_k (Fig. 1.1), i.e., $\rho(i, j, k)$ corresponds to a permutation

$$\left(\begin{array}{l} 1 \ \ldots \ i-1 \ \boxed{\textbf{i} \ \ \textbf{i+1} \ \ldots \ \ldots \ \ldots \ \textbf{j-2} \ \textbf{j-1}} \ \boxed{\text{j} \ \ldots \ \text{k-1}} \ k \ \ldots \ n \\ \\ 1 \ \ldots \ i-1 \ \boxed{\text{j} \ \ldots \ \text{k-1}} \ \boxed{\textbf{i} \ \textbf{i+1} \ \ldots \ \ldots \ \ldots \ \textbf{j-2} \ \textbf{j-1}} \ k \ \ldots \ n \end{array} \right).$$

Clearly, $\pi \cdot \rho(i, j, k)$ has the effect of moving genes $\pi_i, \pi_{i+1}, \ldots, \pi_{j-1}$ to a new location in a genome. Also, note that for $i < j < k$, $\rho(i, j, k)$ has the effect of exchanging blocks $\pi_i, \ldots, \pi_{j-1}$ and $\pi_j, \ldots, \pi_{k-1}$, and $\rho(i, j, k) = \rho(j, k, i)$.

Given permutations π and σ, the *transposition distance problem* is to find a series of transpositions $\rho_1, \rho_2, \ldots, \rho_t$ such that $\pi \cdot \rho_1 \cdot \rho_2, \ldots, \rho_t = \sigma$ and t is minimum. We call t the *transposition distance* between π and σ. Note that transposition distance between π and σ equals the transposition distance between $\sigma^{-1}\pi$ and the *identity* permutation $\imath$. *Sorting π by transpositions* is the problem of finding transposition distance $d(\pi)$ between π and $\imath$. Note that the "biological" definition of transpositions used in this paper is different from the usual "algebraic" definition.

Transpositions generate the *symmetric group* S_n, and we seek a shortest product of *generators* $\rho_1 \cdot \rho_2, \ldots, \rho_t$ that equals $\pi \epsilon S_n$. Even and Goldreich [8] show that,

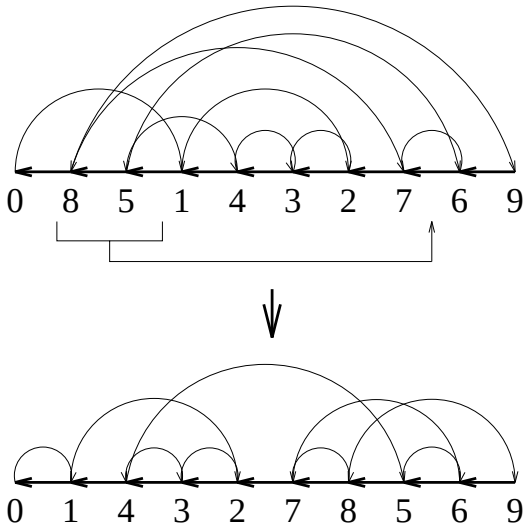


FIG. 1.1. *Transposition* $\rho(1,3,8)$ *on* π *transforms cycle graph* $G(\pi)$ *into* $G(\pi\rho)$.

given a set of generators of a permutation group, determining the shortest product of generators that equals π is NP-hard. In our problem, the generator set is fixed and the complexity status of sorting by transpositions is unknown. The only known polynomially solvable variant of sorting by transpositions is sorting by transpositions $\rho(i, i+1, i+2)$, where the operation is an exchange of adjacent elements. For this problem, polynomial algorithms exist for both linear and circular permutations (Jerrum [15]). Aigner and West [1] found a simple algorithm for sorting by transpositions $\rho(1, 2, i)$ when the operation is reinsertion of the first element.

Sorting by transpositions is a somewhat harder combinatorial problem than the previously studied sorting by reversals; in particular, the transposition diameter of the symmetric group is still unknown. To devise a performance guarantee algorithm for sorting by transpositions, we establish lower bounds for transposition distance based on the notion of the *cycle* graph of a permutation. In section 2 we show that the number of alternating cycles in this edge-colored graph is a bottleneck for sorting by transposition. In section 3 we derive upper bounds for sorting by transposition based on the analysis of *crossing* cycles in the cycle graph. More involved analysis in section 4 provides even better upper bounds in the case where the cycle graph contains *long* cycles. However, this construction breaks for *short* cycles. Somewhat surprisingly, the analysis of *parity* of cycles in the cycle graph provides a compromise and leads to a 1.75 performance guarantee algorithm (section 5). Finally, in section 6 we devise a 1.5 performance guarantee algorithm for sorting by transpositions by exploiting both the structure and parity of crossing cycles in the cycle graph. As an application, we derive a nontrivial upper bound on the transposition diameter of the symmetric group. Algorithms for sorting by reversals and transpositions present the first steps toward the solutions of two open biological problems described in the last section.

2. Lower bounds for sorting by transpositions. For all $0 \le i \le n$, the pair (π_i, π_{i+1}) is a *breakpoint* if $\pi_{i+1} \neq \pi_i + 1$. Observe that the identity permutation is the only permutation with 0 breakpoints, and therefore, sorting a permutation corresponds to decreasing the number of breakpoints. However, this correspondence

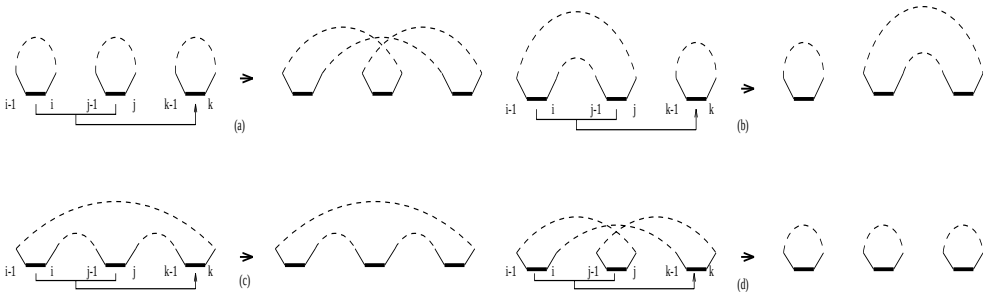


FIG. 2.1. *Transpositions change number of cycles in cycle graphs.*

is not tight in that a permutation with few breakpoints may be more distant from the identity permutation than one with more breakpoints. Also, it is easy to see that a transposition can decrease the number of breakpoints by at most 3, implying a trivial lower bound of $d(\pi) \geq \frac{\#\text{breakpoints}(\pi)}{3}$. However, not all permutations allow transpositions that reduce the number of breakpoints by 3, so the bound is not tight. We introduce the notion of a cycle graph of a permutation and use it to obtain improved lower bounds.

A directed edge-colored *cycle graph* of π, denoted by $G(\pi)$, is the graph with vertex set $\{0, 1, \ldots, n+1\}$ and edge set defined as follows. For all $1 \leq i \leq n+1$, gray edges are directed from $i-1$ to i and black edges from π_i to π_{i-1} (In Fig. 1.1, black edges are shown by thick lines and gray edges are shown by thin lines).

An *alternating cycle* of $G(\pi)$ is a directed cycle in which the edges alternate colors. Observe that for each vertex in $G(\pi)$ every incoming edge is uniquely paired with an outgoing edge of different color. This implies that there is a unique decomposition of the edge set of $G(\pi)$ into alternating cycles. In what follows, we will use *cycle* to refer to an alternating cycle and use *k-cycle* to refer to an alternating cycle of length $2k$. Also, we call a k-cycle *long* if $k > 2$, and *short* otherwise.

There are a total of $2(n+1)$ edges and at most $(n+1)$ cycles in $G(\pi)$ (the identity permutation is the only permutation with $n+1$ cycles). For a permutation π, denote the number of cycles in $G(\pi)$ as $c(\pi)$. Then the sequence of transpositions that sort π must increase the number of cycles from $c(\pi)$ to $n+1$. For a permutation π and a transposition ρ, denote $\Delta c(\rho) = c(\pi\rho) - c(\pi)$ as the change in number of cycles due to transposition ρ.

LEMMA 2.1. $\Delta c(\rho) \in \{2, 0, -2\}$.

Proof. A transposition $\rho(i, j, k)$ involves six vertices of graph $G(\pi)$ $(\pi_{i-1}, \pi_i, \pi_{j-1}, \pi_j, \pi_{k-1}, \pi_k)$ and leads to removing three black edges $((\pi_i, \pi_{i-1}), (\pi_j, \pi_{j-1})$, and $(\pi_k, \pi_{k-1}))$ and adding three new black edges $((\pi_j, \pi_{i-1}), (\pi_i, \pi_{k-1})$, and $(\pi_k, \pi_{j-1}))$.

Three removed edges belong to either three, two, or one cycles in the cycle decomposition of $G(\pi)$. In the case where the removed edges belong to three cycles, $c(\pi\rho) = c(\pi) - 3 + 1$, since these three cycles correspond to one cycle in $G(\pi\rho)$ (Fig. 2.1a). In the case where the removed edges belong to two cycles, $c(\pi\rho) = c(\pi) - 2 + 2$, since these two cycles correspond to two cycles in $G(\pi\rho)$ (Fig. 2.1b). In the case where the removed edges belong to a single cycle C, there are two subcases (Figs. 2.1c and 2.1d). In the subcase shown in Fig. 2.1c, $c(\pi\rho) = c(\pi) - 1 + 1$, since C corresponds to one cycle in $G(\pi\rho)$. In the subcase shown in Fig. 2.1d, $c(\pi\rho) = c(\pi) - 1 + 3$, since C corresponds to three cycles in $G(\pi\rho)$. □

Lemma 2.1 immediately gives a lower bound on $d(\pi)$.

THEOREM 2.2. $d(\pi) \geq \frac{n+1-c(\pi)}{2}$.

A cycle in $G(\pi)$ is *odd* if it has an odd number of black edges and *even* otherwise. To establish a better lower bound we analyze odd and even cycles separately. Define $c_{odd}(\pi)$ ($c_{even}(\pi)$) as the number of odd (even) cycles in π. For a permutation π, and a transposition ρ, denote $\Delta c_{odd}(\rho) = c_{odd}(\pi\rho) - c_{odd}(\pi)$ as the change in number of odd cycles due to transposition ρ.

LEMMA 2.3. $\Delta c_{odd}(\rho) \in \{2, 0, -2\}$.

Proof. The proof of Lemma 2.1 implies that the only case when a transposition ρ leads to creating more than two new cycles in $G(\pi\rho)$ is the case presented in Fig. 2.1d. In this case, three cycles are added to $G(\pi)$ and one cycle is removed from $G(\pi)$. If all three added cycles are odd, then the removed cycle is also odd, and $c_{odd}(\pi\rho) = c_{odd}(\pi) - 1 + 3$. Therefore $\Delta c_{odd}(\rho) \leq 2$. This condition, Lemma 2.1, and parity considerations imply $\Delta c_{odd}(\rho) \in \{2, 0, -2\}$. □

As the identity permutation has $n + 1$ odd cycles, Lemma 2.3 implies a better bound.

THEOREM 2.4. $d(\pi) \geq \frac{n+1-c_{odd}(\pi)}{2}$.

Define $d(n) = \max_{\pi \in S_n} d(\pi)$ to be the *transposition diameter* of the symmetric group of order n. Observing that for $\pi = n\ n-1, \ldots, 2\ 1$, $c_{odd}(\pi) = 1$ if n is even and $c_{odd}(\pi) \leq 2$ if n is odd, the transposition diameter of the symmetric group S_n is at least $\lfloor \frac{n}{2} \rfloor$. One can verify that $d(n\ n-1, \ldots, 1) \leq \lfloor \frac{n}{2} \rfloor + 1$ for all n and $d(n) = d(n\ n-1, \ldots, 1) = \lfloor \frac{n}{2} \rfloor + 1$ for $3 \leq n \leq 10$.

3. Upper bounds for sorting by transpositions. For $x \in \{2, 0, -2\}$, define an *x-move* on π as a transposition ρ such that $\Delta c(\rho) = x$. In order to sort faster, we would like to use as many 2-moves as possible. In this section, we study the structure of cycles which allow 2-moves and use that to devise a performance guarantee algorithm for sorting by transpositions.

We number the black edges of the cycle graph $G(\pi)$ from 1 to $n+1$ by assigning label i to a black edge from π_i to π_{i-1}. We say that transposition $\rho(i, j, k)$ *acts* on edges i, j, and k. We also say that a transposition $\rho(i, j, k)$ *acts* on a cycle C if all three black edges i, j, and k belong to C. The proof of Lemma 2.1 implies the following simple observations.

LEMMA 3.1. *If a transposition ρ acts on a cycle and creates more than one new cycle in $G(\pi\rho)$, then ρ is a 2-move.*

LEMMA 3.2. *If a transposition ρ acts on edges belonging to exactly two different cycles, then ρ is a 0-move.*

Figure 2.1 presents two different kinds of cycles—*nonoriented* for which no 2-moves are possible (Fig. 2.1c) and *oriented* for which a 2-move is possible (Fig. 2.1d). Below we give a formal definition of oriented and nonoriented cycles.

Consider a k-cycle C visiting (in order) the black edges $i_1, \ldots, i_k$. A cycle C can be written in k possible ways depending on the choice of the first black edge. Below we assume that the initial black edge i_1 of cycle C starts at its "rightmost" vertex in π, i.e., $i_1 = \max_{1 \leq t \leq k} i_t$.

For all $k > 1$, a cycle $C = (i_1, \ldots, i_k)$ is *nonoriented* if $i_1, \ldots, i_k$ is a decreasing sequence; otherwise C is an *oriented* cycle. We will also use a characterization of nonoriented cycles in the terms of *edge directions*. A gray edge joining $\pi_t = i - 1$ with $\pi_s = i$ in $G(\pi)$ is directed *left* if $t > s$ and is directed *right* otherwise. Clearly, a cycle $C = (i_1, \ldots, i_k)$ is nonoriented iff $k > 1$ and C has exactly one right edge (a gray edge between black edges i_k and i_1).

LEMMA 3.3. *If C is an oriented cycle, then there exists a 2-move acting on C. If C is a nonoriented cycle, then there exist no 2-moves acting on C.*

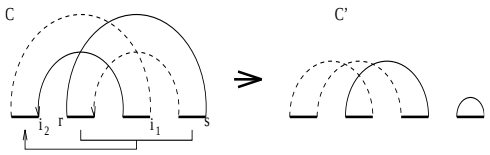


FIG. 3.1. *A 0-move creating an oriented cycle.*

Proof. Let $C = (i_1, \ldots, i_k)$ be an oriented cycle and let $3 \le t \le k$ be an index such that $i_t > i_{t-1}$. Consider a transposition $\rho(i_{t-1}, i_t, i_1)$ acting on C. This transposition creates a 1-cycle (on vertices $\pi_{i_{t-1}-1}$ and π_{i_t}) and some other cycles. Therefore, by Lemma 3.1, ρ is a 2-move. □

Lemmas 3.2 and 3.3 imply the following theorem.

THEOREM 3.4. *For an arbitrary (unsorted) permutation π, there exists either a 2-move or a 0-move followed by a 2-move.*

Proof. If $G(\pi)$ has an oriented cycle then, by Lemma 3.3, a 2-move is possible. Otherwise, consider a nonoriented cycle $C = (i_1, \ldots, i_k)$ and let r be a position of the maximal element of π in the interval $[i_2, i_1 - 1]$. Let s be a position of $\pi_r + 1$ in π (Fig. 3.1). Clearly $s \notin [i_2, i_1]$. Without loss of generality, assume that $s > i_1$, and consider a transposition $\rho(r + 1, s, i_2)$ (Fig. 3.1). The transposition ρ acts on edges of two different cycles; therefore by Lemma 3.2 ρ is a 0-move. Since ρ changes the direction of the left edge (π_{i_1-1}, π_{i_2}), and does not change direction of the right edge (π_{i_k-1}, π_{i_1}), the cycle C' containing these edges in $G(\pi\rho)$ has at least two right edges. Therefore C' is an oriented cycle allowing a 2-move (Lemma 3.3). □

Theorem 3.4 provides an increase of $c(\pi)$ by at least 2 in two consecutive moves and implies the following upper bound for sorting by transpositions.

THEOREM 3.5. *Any permutation π can be sorted in $n + 1 - c(\pi)$ transpositions.*

Theorems 2.2 and 3.5 imply an approximation algorithm for sorting by transpositions with performance guarantee 2. In the following sections, we give a better upper bound by disallowing -2-moves and forcing at least two consecutive 2-moves between any two 0-moves. In our approximation algorithm, we will use only 0- and 2-moves, although we do not have proof that an optimal sequence of transpositions exists which does not use -2-moves.

4. Crossing cycles. Theorem 3.4 shows that the number of 2-moves can be made greater than or equal to the number of 0-moves. In order to improve the performance ratio for sorting by transposition, we need to further increase the number of 2-moves. Theorem 4.7 provides the first step toward such an improvement, but first we need to prove a series of technical lemmas.

Consider a triple of black edges x, y, z belonging to the same cycle C in $G(\pi)$. C induces a cyclic order on x, y, z, and among three possible representations of this order we choose the one starting from the rightmost black edge $\max\{x, y, z\}$ as the canonical representation for a triple (x, y, z). A triple (in a canonical order) is called *nonoriented* if $x > y > z$ and *oriented* otherwise. For example, a triple (k, j, i) in Fig. 2.1c is nonoriented while triple (k, i, j) in Fig. 2.1d is oriented. All triples of a nonoriented cycle are nonoriented. On the other hand, every oriented cycle has at least one oriented triple.

Ordered sequences of integers $\{v_1 < \cdots < v_k\}$ and $\{w_1 < \cdots < w_k\}$ are *interleaving* if either $v_1 < w_1 < v_2 < w_2 < \cdots < v_k < w_k$ or $w_1 < v_1 < w_2 < v_2 \cdots < w_k < v_k$. Sets of integers V and W are interleaving if orderings of V and W are interleaving.

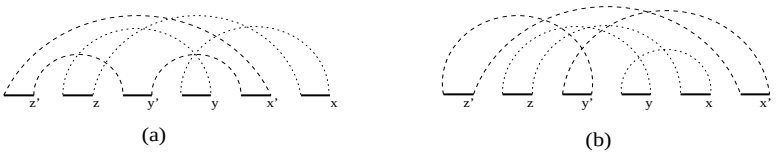


FIG. 4.1. *Crossing and noninterfering cycles.*

Let (x, y, z) be a nonoriented triple, i.e., $x > y > z$. A transposition $\rho(i, j, k)$ is a *shuffling* transposition with respect to a triple (x, y, z) if the sets $\{i, j, k\}$ and $\{x, y, z\}$ interleave.

LEMMA 4.1. *Let (x, y, z) be a triple in a cycle C, and let $i, j, k \notin C$ be black edges in $G(\pi)$. Then $\rho(i, j, k)$ changes the orientation of triple (x, y, z) (i.e., it transforms oriented triple into non-oriented and vice versa) iff ρ is a shuffling transposition for (x, y, z).*

LEMMA 4.2. *If C is nonoriented, then for all triples $(x, y, z) \in C$, transposition $\rho(z, y, x) = \rho(y, x, z)$ transforms C into a nonoriented cycle in $G(\pi\rho)$.*

We will also need the following lemma specifying some 2-moves acting on oriented cycles.

LEMMA 4.3. *If (x, y, z) is an oriented triple, then $\rho(y, z, x) = \rho(z, x, y)$ is a 2-move.*

Cycles C and C' are *crossing* if there exists an oriented triple in C and a nonoriented triple in C' that are interleaving (Fig. 4.1a). Cycles C and C' are *non-interfering* if there exist oriented triples in C and C' that are not interleaving (Fig. 4.1b).

LEMMA 4.4. *If permutation π has crossing or noninterfering cycles, then there exist two consecutive 2-moves in π.*

Proof. If cycles C and C' in $G(\pi)$ are crossing, there exist an oriented triple $(x, z, y) \in C$ and a nonoriented triple $(x', y', z') \in C'$ which are interleaving (Fig. 4.1a). By Lemma 4.3, a transposition $\rho(z, y, x)$ defines a 2-move on C. On the other hand, since (x, y, z) and (x', y', z') are interleaving, $\rho(z, y, x)$ is a shuffling transposition with respect to (x', y', z'). Thus, by Lemma 4.1 ρ transforms C' into an oriented cycle in $G(\pi\rho)$ and by Lemma 3.3 provides a second 2-move.

Alternatively, if C and C' are noninterfering, then there exist oriented triples $(x, z, y) \in C$ and $(x', z', y') \in C'$ which are noninterleaving (Fig. 4.1b). By Lemma 4.3, a transposition $\rho(z, y, x)$ defines a 2-move on C. Furthermore, (x', z', y') remains an oriented triple (Lemma 4.1) of C' in $G(\pi\rho)$, which provides a second 2-move. □

We say that a transposition *acts* on two cycles C and C' in $G(\pi)$ if it acts on black edges of both C and C'. To prove Theorem 4.7 below, we will need the following observation about transpositions acting on two cycles.

LEMMA 4.5. *Let C be a cycle containing black edges x and y and let D be a cycle containing black edges x' and y'. Let ρ be a transposition acting on three of four black edges x, y, x', y'.*

- *If $\{x, y\}$ does not interleave with $\{x', y'\}$, then ρ creates a cycle with a nonoriented triple.*
- *If $\{x, y\}$ interleaves with $\{x', y'\}$, then ρ creates a cycle with an oriented triple.*

Proof. See Fig. 4.2. All other cases are symmetric. □

We say that cycle $C = (i_1, \ldots, i_k)$ *spans* cycle $D = (j_1, \ldots, j_l)$, if $i_k < j_l < j_1 < i_1$. The following lemma illustrates an important property of nonoriented cycles.

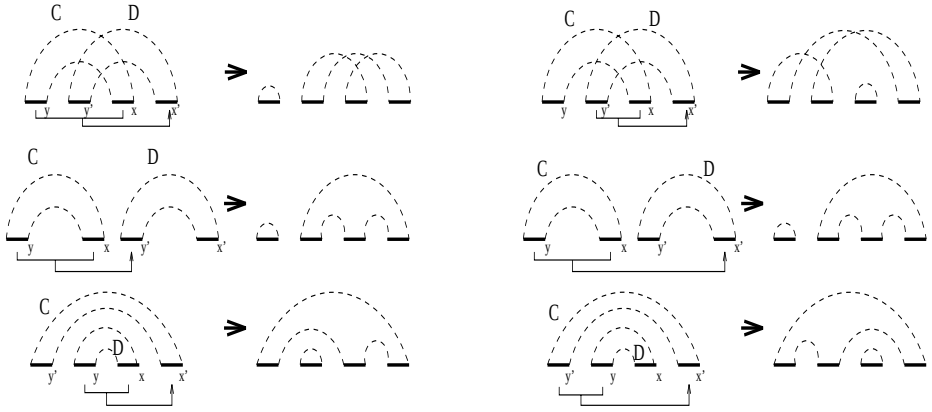


FIG. 4.2. *Transpositions acting on two cycles.*

LEMMA 4.6. *For every nonoriented cycle $C = (\dots a, \dots, b \dots)$, with arbitrary edges a, b, there exists a cycle $D(\dots, c \dots, d \dots)$ such that (a, b) and (c, d) interleave.*

Proof. Let $\pi_c = \max_{i \in [b, a-1]} \pi_i$ and $\pi_d = \pi_c + 1$. Choice of c implies that $d \notin [b, a-1]$, as C is nonoriented $d \neq a$, implying that $d \notin [b, a]$. Therefore, (c, d) and (a, b) interleave. □

THEOREM 4.7. *If there exists a long cycle in $G(\pi)$, then either a 2-move or a 0-move followed by two consecutive 2-moves is possible in π.*

Proof. If $G(\pi)$ has an oriented cycle, then by Lemma 3.3 a 2-move is possible. Also, if there exist nonoriented long cycles C and D with interleaving triples $(r, s, t) \in C$ and $(x, y, z) \in D$, then a 0-move ρ acting on edges z, y, x is a shuffling transposition for C. By Lemma 4.1, ρ transforms C into an oriented cycle C'. By Lemma 4.2 ρ transforms D into a nonoriented cycle D'. It is easy to see that C' and D' are crossing; therefore, by Lemma 4.4 there exist two consecutive 2-moves in $G(\pi\rho)$.

Therefore, assume that no two cycles have interleaving triples. Pick a nonoriented long cycle $C = (i_1, \dots, i_k)$, such that C is not spanned by any long cycle. Find a cycle $D = (x, \dots, c, \dots, d, \dots, y)$ such that the pairs (c, d) and (i_1, i_k) interleave (Lemma 4.6). Note that if $y < i_k$, then $x < i_1$; otherwise D would span C. On the other hand, if $y > i_k$, then $x > i_1$; otherwise (c, d) and (i_1, i_k) would not interleave. Therefore, either $y < i_k < x < i_1$ or $i_k < y < i_1 < x$. Without loss of generality, we assume the latter. Let s be the rightmost edge in C to the left of y, i.e., $s = \max_{i \in C, i < y} i$. Two cases arise.

$s > i_k$: Find cycle $E = (v, \dots, c, \dots, d, \dots, u)$ such that the pairs (c, d) and (i_k, s) interleave (Lemma 4.6). If $u < i_k$, then $v < s$ because, otherwise, E either spans C ($v > i_1$) or has an interleaving triple with $(i_k, s, i_1) \in C$ ($s < v < i_1$). If $u > i_k$ (Fig. 4.3a), then four cases arise depending on v lying in one of the intervals $[s, y], [y, i_1], [i_1, x]$ or $[x, n+1]$ (Fig. 4.3b-e). The transpositions $\rho(x, y, v)$ in Fig. 4.3a and $\rho(x, y, u)$ in Figs. 4.3b-4.3e are shuffling w.r.t. the triple (i_1, s, i_k) of C, and by Lemma 4.1 transform C into an oriented cycle C' in $G(\pi\rho)$. ρ also transforms D and E into D' and a 1-cycle in $G(\pi\rho)$. From Lemma 4.5, D' is oriented in Fig. 4.3a. In the remaining cases, D' is oriented when $v \in [y, x-1]$ and nonoriented otherwise (Lemma 4.5). Observe that in the first case C' and D' are crossing (Figs. 4.3b, 4.3e); otherwise they are noninterfering (Figs. 4.3c, 4.3d). In either case, two 2-moves are possible in $G(\pi\rho)$(Lemma 4.4).

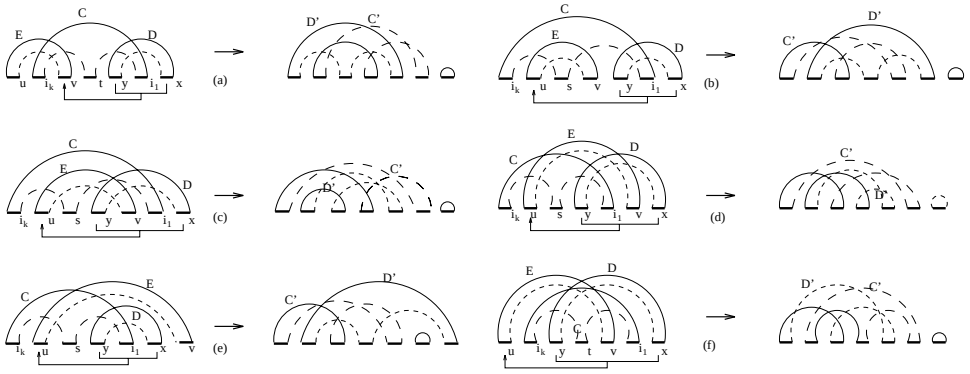


FIG. 4.3. 0-*move leading to two* 2-*moves.*

Algorithm *Tsort(π)*

1. While $G(\pi)$ has a long cycle, perform either a 2-move or a $0, 2, 2$-move (Theorem 4.7).
2. If $G(\pi)$ has only short cycles, perform a 0-move followed by a 2-move (Theorem 3.4).

FIG. 5.1. *Algorithm* Tsort *for sorting by transpositions.*

$s = i_k$: Let t be the leftmost black edge in C to the right of y, i.e., $t = \min_{i \in C, i > y} i$. As C is a long cycle, $t < i_1$. Find $E = (v, \ldots, c, \ldots, d, \ldots, u)$ such that the pairs (c, d) and (t, i_1) interleave (Lemma 4.6). Cycle E is different from cycle D because, otherwise, E and C would have interleaving triples. If $v > i_1$, then $u > t$ because, otherwise, E either spans C ($u < i_1$) or has an interleaving triple with $(i_1, t, i_k) \in C$ ($i_1 < u < t$). This case is similar to the cases shown in Figs. 4.3d, 4.3e. If $v < i_1$, then three cases arise depending on which of the intervals $[0, i_k], [i_k, y]$, or $[y, t]$ contains u. The first of these cases is shown in Fig. 4.3f, while the other two are symmetric to cases in Fig. 4.3c and 4.3e, respectively. In Fig. 4.3f, the transposition $\rho(x, y, u)$ transforms C into a nonoriented cycle C' (Lemma 4.1), and transforms cycles D, E into an oriented cycle D' and a 1-cycle in $G(\pi\rho)$ (Lemma 4.5). Further, C' and D' are crossing, and therefore two 2-moves are possible in $G(\pi\rho)$. □

5. Mixing odd and even cycles. Theorem 4.7 guarantees creating at least four cycles in three moves, thus providing $\Delta c(\rho) = \frac{4}{3}$ on average, which is better than $\Delta c(\rho) = 1$, given by Theorem 3.4. However, it can be applied only when $G(\pi)$ has long cycles. In case $G(\pi)$ only has short cycles, the best we can guarantee is a 0-move followed by a 2-move creating four 1-cycles from two 2-cycles (Theorem 3.4). Theorems 3.4 and 4.7 motivate the algorithm *Tsort* (Fig. 5.1).

Does *Tsort* achieve a performance ratio of better than 2? Unfortunately, in the case that $G(\pi)$ has only short cycles, the 0-move followed by a 2-move provides only $\Delta c(\rho) = \frac{4-2}{2} = 1$ on average. However, for these two moves, $\Delta c_{odd}(\rho) = \frac{4-0}{2} = 2$, thus achieving a maximal rate of creating odd cycles from the perspective of Theorem 2.4. On the other hand, Theorem 4.7 does not guarantee yet that $\Delta c_{odd}(\rho) = 2$ for every 2-move. Therefore, if we use either the number of cycles or the number of odd cycles

as our objective function, we cannot guarantee a performance ratio better than 2. Somewhat surprisingly, we show that a *mixed* objective function which gives different weights to odd and even cycles leads to an improved performance guarantee.

THEOREM 5.1. Tsort *provides a performance guarantee of* 1.75 *for sorting by transpositions.*

Proof. For arbitrary $x \geq 1$, define the objective function $f(\pi) = xc_{odd}(\pi) + c_{even}(\pi)$, where $c_{odd}(\pi)$ and $c_{even}(\pi)$ are the number of odd and even cycles in $G(\pi)$, respectively. Clearly, for this range of x, $f(\pi)$ is uniquely maximized by the identity permutation, and sorting a permutation corresponds to maximizing f. Observe that the maximum gain any transposition ρ can achieve is $\Delta f(\rho) = f(\pi\rho) - f(\pi) = 2x$. We now evaluate the maximum Δf guaranteed by Theorems 3.4 and 4.7.

In the case that $G(\pi)$ only has short cycles, Theorem 3.4 guarantees that in two moves, four 1-cycles are created from two 2-cycles, implying a gain of $4x-2$ over two moves, or an average gain of $2x-1$ in one transposition. In any 2-move, two new cycles are created and, in the worst case (if both are even) we can still guarantee a gain of 2. By construction, a 0-move in Theorem 4.7 either creates a 1-cycle or does not change the number of black edges in any cycle. Therefore $\Delta f \geq 0$ for any 0-move. Moreover, Theorem 4.7 guarantees that any such 0-move is followed by two 2-moves, implying an average gain of $\frac{4}{3}$. It follows that $\Delta f \geq \min\{\frac{4}{3}, 2x-1\}$ on the average. Comparing the best possible gain of $2x$ against the gain provided by *Tsort*, we get a performance guarantee of

$$\frac{2x}{\min\{\frac{4}{3}, 2x-1\}}.$$

The best performance is achieved for $x = \frac{7}{6}$, resulting in the approximation ratio 1.75. □

6. A 1.5 approximation algorithm for sorting by transposition. In order to improve performance still further, we need to strengthen Theorem 4.7. Note that Theorem 4.7 only guarantees an increase in the number of cycles. However, the identity permutation has $n+1$ cycles, all of length one, indicating that we need to increase the number of odd cycles. By choosing appropriate 2-moves, we shall ensure that the number of odd cycles increases by at least two in every 2-move.

We call a transposition ρ *valid* if $\Delta c(\rho) = \Delta c_{odd}(\rho)$. For a cycle C containing edges i and j, define $d(i,j)$ as the number of black edges between vertices π_i and π_j in C (in particular, $d(i,j) = 1$ for consecutive black edges i and j).

LEMMA 6.1. *If there exists an oriented cycle in* $G(\pi)$*, then either a valid* 2*-move or a valid* 0*-move followed by two consecutive valid* 2*-moves is possible in* π*.*

Proof. Suppose there is no valid 2-move in π. For an arbitrary oriented cycle C in $G(\pi)$, consider the following set S of oriented triples of C such that the distance between the first and second elements of the triple is odd:

$$S = \{(x,y,z) : \ x,y,z \in C \text{ and } d(x,y) \text{ is odd}\}.$$

The observation that every oriented cycle C has an oriented triple (x,y,z) such that x and y are the consecutive black edges in C implies that S is nonempty. Let (x,y,z) be a triple in S with maximal x.

A transposition ρ acting on edges y, z, and x transforms C into three cycles C_1, C_2, and C_3 consisting of $d(x,y), d(y,z)$, and $d(z,x)$ black edges. As $(x,y,z) \in S$, cycle C_1 is odd. If either C_2 or C_3 is odd, then $\Delta c_{odd}(\rho) = 2$ and ρ is a valid 2-move, contradicting the assumption that there are no valid 2-moves in π. Therefore both

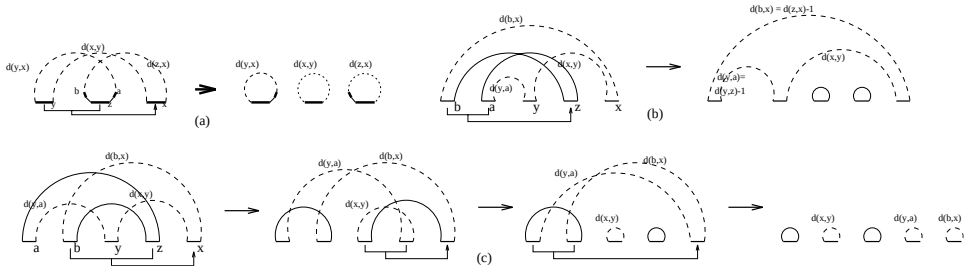


FIG. 6.1. *Valid* 2*-moves and* $0, 2, 2$*-moves on an oriented cycle.*

$d(y, z)$ and $d(z, x)$ are even. As both $d(y, z)$ and $d(z, x)$ are even, the fragments of C from y to z and from z to x contain at least two edges. Let a be a black edge preceding z in C and b be a black edge following z in C (Fig. 6.1a).

If $y < a < x$, then transposition ρ acting on edges y, a, and x creates cycles of length $d(y, z) - 1$ and $d(x, y)$. Both these numbers are odd and, therefore, ρ is a valid 2-move, thus contradicting the assumption. Therefore $a \notin [y, x]$. Symmetric arguments demonstrate that $b \notin [y, x]$.

If $a > x$, then (a, z, x) is an oriented triple with odd $d(a, z) = 1$, thus contradicting the choice of (x, y, z). Therefore $a < y$. If $b > x$, then (b, a, z) is an oriented triple with odd $d(b, a) = d(b, x) + d(x, y) + d(y, a) = (d(z, x) - 1) + d(x, y) + (d(y, z) - 1)$, thus contradicting the choice of (x, y, z). Therefore $a, b < y$.

The situations described by conditions $b < a$ and $a < b$ are presented in Figs. 6.1b and 6.1c. If $b < a$, then $\rho(b, a, z)$ is a valid 2-move (Fig. 6.1b). If $a < b$, then there exist 2-moves but no valid 2-moves in π. However, there exists a valid 0-move followed by two consecutive valid 2-moves (Fig. 6.1c). □

Fig. 6.1c presents an example of an oriented cycle which does not allow valid 2-moves. This cycle has a complicated "self-interleaving" structure and, in the following, we try to avoid creating such cycles. In order to achieve this goal, we define *strongly oriented* cycles, which have the simplest "self-interleaving" structure among all oriented cycles.

Let $C = (i_1, \ldots, i_k)$ be a cycle in $G(\pi)$ and let $C^* = (i_1 = j_1 > \cdots > j_k)$ be a sequence of black edges of C in decreasing order. Sequences C and C^* coincide for a nonoriented cycle and are different otherwise. Define *strongly oriented* cycles as oriented cycles for which C^* can be transformed into C by a single transposition, i.e., C can be partitioned into strips $C_1 = (i_1, \ldots, i_a)$, $C_2 = (i_{a+1}, \ldots, i_b)$, $C_3 = (i_{b+1}, \ldots, i_c)$, and $C_4 = (i_{c+1}, \ldots, i_k)$ such that $C = C_1C_2C_3C_4$ and $C^* = C_1C_3C_2C_4$ (C_4 might be empty). For example, Fig. 6.1b gives an example of a strongly oriented cycle, as $C = xyabz$ is transformed into $C^* = xzyab$ by a single transposition. Clearly, every strongly oriented cycle has exactly two right edges. On the other hand, not every oriented cycle with two right edges is strongly oriented (Fig. 6.1c).

LEMMA 6.2. *A strongly oriented cycle allows a valid* 2*-move.*

Proof. Depending on whether or not C_4 is empty, there are two kinds of cycles, as shown in Fig. 6.2, with $left$+mid+$right$ black edges (in Fig. 6.2c, $mid = mid'$+mid''). Dashed lines in the figure represent alternating paths of zero or more edges. In the following, we shall abuse notation by referring to both the sets of edges and their numbers as $left, mid$, and $right$.

In Fig. 6.2a, consider transpositions of the form $\rho(i, j, k)$, where i is the leftmost mid edge, j is the rightmost $right$ edge, and k is a $left$ edge. As all such triples (i, j, k) are oriented; $\rho(i, j, k)$ is a 2-move.

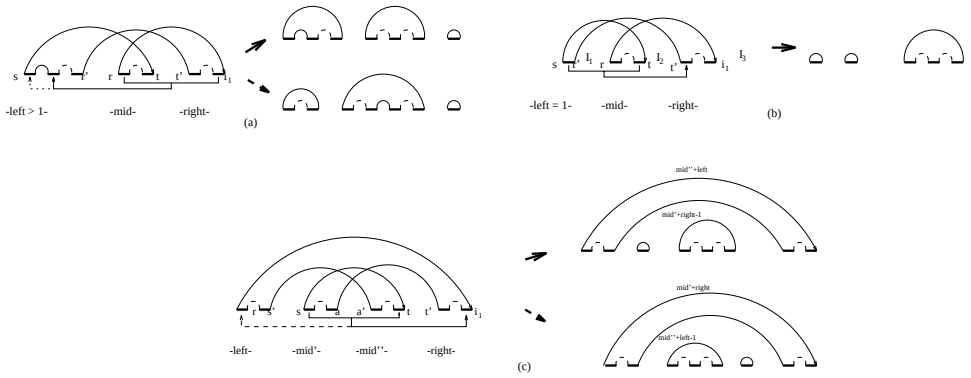


FIG. 6.2. *Strongly oriented cycles:* (a), (b) *First kind.* (c) *Second kind.*

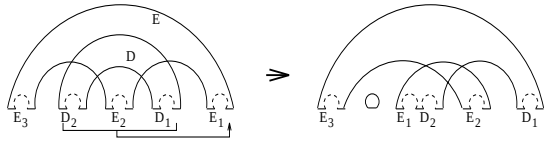


FIG. 6.3. *Transforming two nonoriented cycles into a strongly oriented cycle.*

Figure 6.2a corresponds to the case $left > 1$ and presents two such transpositions, say, $\rho_1(i, j, k_1)$ and $\rho_2(i, j, k_2)$, in which k_1 and k_2 are the two leftmost *left* edges. Both ρ_1 and ρ_2 are 2-moves and create three cycles. One of these cycles is a 1-cycle. If $left > 1$, then an appropriate choice of either ρ_1 or ρ_2 provides at least one more odd cycle, thus indicating that the chosen transposition is a valid 2-move. If $left = 1$, then the transposition ρ shown in Fig. 6.2b creates at least two 1-cycles, thus indicating that ρ is a valid 2-move.

In Fig. 6.2c, the transposition ρ inserting a "middle interval" into the leftmost edge creates cycles of length $1, mid'' + left - 1$ and $mid' + right$. On the other hand, a transposition ϱ inserting a middle interval into the rightmost edge creates cycles of length $1, mid'' + left$ and $mid' + right - 1$. Therefore, either ρ or ϱ creates at least two odd cycles, thus ensuring a valid 2-move in π. □

Next, we present two lemmas which show how strongly oriented cycles arise from nonoriented cycles.

LEMMA 6.3. *If ρ is a shuffling transposition on a nonoriented cycle C, then ρ transforms C into a strongly oriented cycle in $G(\pi\rho)$.*

Proof. The proof follows from the definition. □

LEMMA 6.4. *Let $D(x, \ldots, y)$ and $E(x', \ldots, y')$ be two nonoriented cycles in $G(\pi)$ with no interleaving triples, and let ρ be a transposition acting on three of four black edges x, y, x', y'. Then ρ creates a strongly oriented cycle iff D and E have interleaving pairs of edges.*

Proof. Figure 6.3 presents cycles D and E with interleaving pairs of edges, but no interleaving triple. Assume w.l.o.g that the edges of D partition E into three strips $E = E_1E_2E_3$ (E_3 is possibly empty), while the edges of E partition the edges of D into two $D = D_1D_2$. The transposition ρ transforms D and E into a 1-cycle and a cycle F visiting (in order) edges $D_1D_2E_1E_2E_3$. On the other hand, $F^* = D_1E_2D_2E_1E_3$ which can clearly be transformed into F by a transposition.

If D and E have no interleaving pairs of edges, then it is easy to verify that ρ transforms D and E into a 1-cycle and a nonoriented cycle F. □

Every strongly oriented cycle has exactly two right edges, one of which is of the form (r, i_1). Label the other as (s, t). For strongly oriented cycles of the first kind (Fig. 6.2a), define

$$r' = \max_{i \in left} i \text{ and } t' = \min_{i \in right} i,$$

and consider three intervals $I_1(C) = [r', r], I_2(C) = [t, t']$, and $I_3 = [0, s] \cup [i_1, n+1]$. For strongly oriented cycles of the second kind (Fig. 6.2c), define

$$s' = \max_{i \in left} i \ , \ t' = \min_{i \in right} i \ , \ a = \max_{i \in mid'} i \text{ and } a' = \min_{i \in mid''} i,$$

and consider intervals $I_1(C) = [s', s], I_2(C) = [t, t']$, and $I_3(C) = [a, a']$.

A strongly oriented cycle C and a nonoriented cycle $C' = (i_1, \ldots, i_k)$ are *strongly crossing* if there exists a black edge x in C' such that each of the sets $I_1(C), I_2(C)$, and $I_3(C)$ contains exactly one element of the triple (i_1, x, i_k). Note that 2-moves for C described in the proof of Lemma 6.2 form shuffling transpositions w.r.t. (i_1, x, i_k). This observation and Lemma 6.2 imply the following.

LEMMA 6.5. *If $G(\pi)$ has strongly crossing cycles, then there exist two consecutive valid 2-moves in $G(\pi)$.*

Next, we modify the concept of "noninterfering" cycles after which we shall have all the tools needed to strengthen Theorem 4.7. A transposition ρ is *safe*, with respect to a strongly oriented cycle $C \in G(\pi)$, if it transforms C into a strongly oriented cycle in $G(\pi\rho)$. The following lemma gives a sufficient condition for a transposition to be safe.

LEMMA 6.6. *Let C be a strongly oriented cycle, and let $(x, y, z) \notin C$ be a triple such that no edge of C lies in the region between x and y. Then, a transposition acting on (x, y, z) is safe w.r.t. C.*

Let cycles C and C' be strongly oriented. C is *strongly noninterfering* w.r.t. C' if it has a right edge (a, b) such that no black edge of C' lies in the interval $[a, b]$.

LEMMA 6.7. *If $G(\pi)$ has strongly noninterfering cycles, then there exist two consecutive valid 2-moves in $G(\pi)$.*

Proof. Let C be strongly noninterfering w.r.t. C'. Consider a valid 2-move $\rho(x, y, z)$ on C described in the proof of Lemma 6.2. Observe that one of the right edges in C is of the form (x, y) and therefore includes the region $[x, y]$, and the other right edge includes the interval $[y, z]$. Therefore, if C is strongly noninterfering w.r.t. C', then either no black edge of C' lies in $[x, y]$ or no black edge of C' lies in $[y, z]$. In either case, $\rho(x, y, z)$ is safe w.r.t. C' (Lemma 6.6). This implies that a valid 2-move on C' follows a valid 2-move on C'. □

Finally, we can prove a stronger version of Theorem 4.7.

THEOREM 6.8. *If there exists a long cycle in $G(\pi)$, then either a valid 2-move or a valid 0-move followed by two consecutive valid 2-moves is possible in π.*

Proof. We mimic the proof of Theorem 4.7, ensuring that all moves are valid ones.

If $G(\pi)$ has an oriented cycle, then from Lemma 6.1, a valid 2-move or a valid 0-move followed by two valid 2-moves is always possible.

Next, consider the case when $G(\pi)$ has nonoriented cycles C and D with interleaving triples $(r, s, t) \in C$ and $(x, y, z) \in D$. Then, $\rho(x, y, z)$ transforms C into a strongly oriented cycle C' in $G(\pi\rho)$ (Lemma 6.2) and transforms D into a nonoriented cycle D' in $G(\pi\rho)$ (Lemma 4.1). Further observe that each of the intervals $I_1(C'), I_2(C')$,

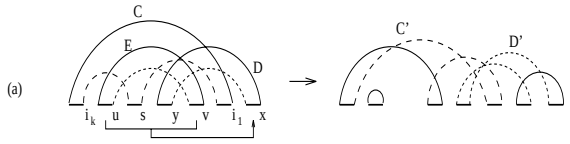


FIG. 6.4. *Transforming C, D, and E into strongly noninterfering cycles.*

and $I_3(C')$ contains exactly one element of a (nonoriented) triple in D'. Therefore, C' and D' are strongly crossing, and from Lemma 6.5, two valid 2-moves are possible in $G(\pi\rho)$.

Therefore, we can assume that $G(\pi)$ has no oriented cycles or cycles with interleaving triples. The proof of theorem holds and we consider them in the following case by case fashion:

Fig. 4.3a. The valid 0-move $\rho(y, x, v)$ transforms D and E into a nonoriented cycle D' (Lemma 6.4) and transforms C into a strongly oriented cycle C' (Lemma 6.2) in $G(\pi\rho)$. Further observe that vertices π_x, π_v, π_y all belong to D' and $\pi_y \in I_1(C')$, $\pi_v \in I_2(C')$, $\pi_x \in I_3(C')$, thereby implying that C' and D' are strongly crossing. From Lemma 6.5, two valid 2-moves are possible in $G(\pi\rho)$.

Fig. 4.3b. The valid 0-move $\rho(y, x, u)$ transforms D and E into a nonoriented cycle D' (Lemma 6.4), and transforms C into a strongly oriented cycle C' (Lemma 6.2) in $G(\pi\rho)$. Observe that $\pi_y \in I_1(C')$ and $\pi_u \in I_2(C')$. Moreover, the choice of s as the rightmost edge to the left of y ensures that there is no edge of C between v and y, and therefore $\pi_v \in I_3(C')$. As vertices π_x, π_v, π_y all belong to D', cycles C' and D' are strongly crossing. From Lemma 6.5, two valid 2-moves are possible in $G(\pi\rho)$.

Fig. 4.3c. In this case, we consider the valid 0-move $\rho(u, v, x)$ (Fig. 6.4) instead of $\rho(y, x, u)$. ρ transforms C into a strongly oriented cycle C' (Lemma 6.2), and transforms D and E into strongly oriented cycle D' (as D and E have no interleaving triples, Lemma 6.4 applies). Define a as the rightmost edge in D to the left of i_1, i.e., $a = \max_{i \in D, i < i_1} i$, and define b as the leftmost edge in C to the right of y, i.e., $b = \min_{i \in C, i > y} i$. Note that $a < b$ because, otherwise, $(i_k, b, i_1) \in C$ and $(y, a, x) \in D$ are interleaving triples. If $b > v$, then there is no edge of C in the interval $[y, v]$, and it follows that C' has no black edge in the region covered by the right edge $(\pi_{y-1}, \pi_x) \in D'$. Therefore D' is strongly noninterfering w.r.t. C'. If $a < b < v$, then there is no black edge of D in the interval $[v, i_1]$, and correspondingly, D' has no black edge in the region covered by the right edge $(\pi_{i_k - 1}, \pi_{i_1}) \in C'$. Therefore, C' is strongly noninterfering w.r.t. D'. In either case, Lemma 6.7 implies that two valid 2-moves are possible in $G(\pi\rho)$.

Fig. 4.3d. The valid 0-move $\rho(x, y, u)$ transforms D and E into strongly oriented cycle D' (as D and E have no interleaving triples, Lemma 6.4 applies) and also transforms C into strongly oriented cycle C' (Lemma 6.2). Let a be the rightmost edge of E to the left of s, and let b the leftmost edge of E to the right of i_1. Note that C has no edge in the region between edges $b \in E$ and $x \in D$ in $G(\pi)$ as $i_1 < b < x$. Also, C has no edge e in the region $[u, a]$ because, otherwise, $(u, a, v) \in E$ would interleave $(e, s, i_1) \in C$. Correspondingly in $G(\pi\rho)$, C' does not have any black edge in the region covered by the right edge $(\pi_{b-1}, \pi_a) \in D'$, implying that D' is strongly noninterfering w.r.t. C'. From Lemma 6.7, two consecutive valid 2-moves are possible in $G(\pi\rho)$.

Algorithm *TransSort(π)*

1. While $G(\pi)$ has a long cycle, perform a valid 2-move or a valid $0, 2, 2$-move (Theorem 6.8).
2. If $G(\pi)$ has only short cycles, perform a good 0-move followed by a valid 2-move (Theorem 6.9).

FIG. 6.5. *Algorithm* TransSort *for sorting by transpositions.*

Fig. 4.3e. The valid 0-move $\rho(x, y, u)$ transforms C into a strongly oriented cycle C' (Lemma 6.2), and cycles D and E into D'. From Lemma 6.4, D' is strongly oriented if D and E have interleaving pairs; otherwise it is nonoriented. In the first case, we use an argument similar to the case in Fig. 4.3d. If D' is nonoriented, then observe that π_y, π_u, π_v all belong to D' and $\pi_y \in I_1(C')$, $\pi_u \in I_2(C')$, and $\pi_v \in I_3(C')$, implying that D' and C' are strongly crossing. From Lemma 6.5, two valid 2-moves are possible in $G(\pi\rho)$.

Fig. 4.3f. The valid 0-move $\rho(x, y, u)$ transforms D and E into strongly oriented cycle D' (as D and E have no interleaving triples, Lemma 6.4 applies) and transforms C into nonoriented C' (Lemma 4.1). Furthermore, $\pi_{i_1}, \pi_t, \pi_{i_k} \in C'$ lie in the regions $I_2(D'), I_1(D')$, and $I_3(D')$, respectively. Therefore, C' and D' are strongly crossing. From Lemma 6.5, two valid 2-moves are possible in $G(\pi\rho)$. □

Theorem 6.8 describes how we can handle the case when $G(\pi)$ has long cycles. For short cycles, we need to formalize the intuitive idea described earlier. Define a 0-move as *good* if it increases the number of odd cycles by two.

THEOREM 6.9. *If $G(\pi)$ has only short cycles, a good* 0*-move followed by a valid* 2*-move is possible.*

Proof. We mimic the proof of Theorem 3.4. The 0-move takes two cycles of length 2 and creates an oriented cycle of length 3 and a cycle of length 1. A valid 2-move is now possible. □

Our proofs are constructive and immediately imply an $O(n^2)$ algorithm *TransSort* for sorting by transpositions. Finally, Theorems 2.4, 6.8, and 6.9 imply the following.

COROLLARY 6.10. *Algorithm* TransSort *sorts permutation π in no more than $\frac{3}{4} \cdot (n + 1 - c_{odd}(\pi))$ transpositions, thereby ensuring a performance guarantee of* 1.5.

COROLLARY 6.11. *The transposition diameter of the symmetric group S_n is at most $\frac{3}{4}n$.*

7. Open problems. Recent advances in large-scale comparative genetic mapping offer exciting prospects for understanding mammalian genome evolution. The large number of conserved segments in the maps of man and mouse suggest that multiple chromosomal rearrangements have occurred since the divergence of lineages leading to humans and mice. In their pioneering paper, Nadeau and Taylor [21] estimated that just 178 ± 39 rearrangements have occurred since this divergence. This estimate survived a ten-fold increase in the amount of the comparative man/mouse mapping information; the new estimate, based on the latest data (Copeland et al. [5]), almost did not change compared to Nadeau and Taylor [21]. However, the arguments used by Nadeau and Taylor [21] are nonconstructive and do not provide any solution to an open biological problem of reconstructing an evolutionary scenario explaining man and mouse genome rearrangements.

Chromosomal rearrangements include not only inversions and transpositions but *translocations, fussions, fissions, insertions, and deletions* as well. A combinatorial analysis of all such rearrangements to derive a scenario of mammalian evolution is far beyond the possibilities of current algorithms. However, some limited applications of algorithms for inversions and transpositions to study chromosome evolutions are already possible. In particular, extreme conservation of genes on X chromosome across mammalian species provides an opportunity to study evolutionary history of X chromosome independently of the rest of the genomes, thus reducing the computational complexity of the problem. According to Ohno's law (Ohno [22]), gene content of X chromosome is assumed to have remained the same throughout mammalian development for the last 125 million years. However, the order of genes on X chromosome has been disrupted several times. The conservative gene content of X chromosome implies that the only translocations which affected the gene order in X chromosome were translocations between two copies of X chromosome and thus might be ignored for our purposes. A recently discovered violation of the Ohno law by the *Csfgmra* gene (Disteche et al. [7]) does not affect this conclusion, since this gene is located at the very end of the human X chromosome. Davisson [6] and Lyon [19] suggested two conflicting scenarios of rearrangements in X chromosome under the assumption that X chromosome was not involved in translocations. Based on the analysis of the latest data on comparative man/mouse mapping, Bafna and Pevzner [3] found the most parsimonious scenario for evolutionary history of X chromosome and corrected the previously suggested scenarios.

Another open problem on genome rearrangements is related to viral evolution. As was mentioned in the introduction, herpes viruses present a particularly hard case for classical sequence comparison. On the other hand, they present a particularly suitable test case for the study of genome rearrangements, since complete sequences of seven diverse herpes viruses are known. Herpes virus genomes contain from 70 to about 200 genes. Detailed comparison of amino acid sequences of viral proteins resulted in an "alphabet" of about 30 conserved genes which were rearranged in different herpes viruses (Hannenhalli et al. [12]). Three types of arrangements of conserved genes exist, corresponding to the α, β, and γ divisions of herpes viruses. Derived lower bounds for the pairwise genome rearrangements of viral genomes allowed us to construct the most parsimonious scenarios for herpes virus evolution. Moreover, there are only three alternative, equally parsimonious, scenarios of genome rearrangements in herpes viruses with three different Steiner points (Hannenhalli et al. [12]). It is impossible to delineate the true scenario among these three based on the currently available data. However, ongoing efforts to map and sequence different herpes virus genomes provide a warrant that a true evolutionary scenario will be found in the future.

REFERENCES

[1] M. Aigner and D. B. West, *Sorting by insertion of leading element*, J. Combin. Theory, 45 (1987), pp. 306–309.

[2] V. Bafna and P. Pevzner, *Genome rearrangements and sorting by reversals*, SIAM J. Comput., 25 (1996), pp. 272–289.

[3] V. Bafna and P. Pevzner, *Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of X chromosome*, Molecular Biology and Evolution, 12 (1995), pp. 239–246.

[4] D. Cohen and M. Blum, *Improved bounds for sorting pancakes under a conjecture*, manuscript, 1993.

[5] N. G. Copeland, N. A. Jenkins, D. J. Gilbert, J. T. Eppig, L. J. Maltals, J. C. Miller, W. F. Dietrich, A. Weaver, S. E. Lincoln, R. G. Steen, L. D. Steen, J. H. Nadeau, and E. S. Lander, *A genetic linkage map of the mouse: Current applications and future prospects*, Science, 262 (1993), pp. 57–65.

[6] M. Davisson, *X-linked genetic homologies between mouse and man*, Genomics, 1 (1987), pp. 213–227.

[7] C. M. Disteche, C. I. Brannan, A. Larsen, D. A. Adler, D. F. Schorderet, D. Gearing, N. G. Copeland, N. A. Jenkins, and L. S. Park, *The human pesuodoautosomal GM-CSF receptor α subunit gene is autosomal in mouse*, Nature Genetics, 1 (1992), pp. 333–336.

[8] S. Even and O. Goldreich, *The minimum-length generator sequence problem is NP-hard*, J. Algorithms, 2 (1981), pp. 311–313.

[9] N. Franklin, *Conservation of genome form but not sequence in the transcription antitermination determinants of bacteriophages $\lambda, \phi 21$ and $P22$*, J. Molecular Evolution, 181 (1985), pp. 75–84.

[10] W. H. Gates and C. H. Papadimitriou, *Bounds for sorting by prefix reversals*, Discrete Math., 27 (1979), pp. 47–57.

[11] A. M. Griffin and M. E. G. Boursnell, *Analysis of the nucleotide sequence of DNA from the region of the thymidine kinase gene of infectious laryngotracheitis virus: potential evolutionary relationships between the herpesvirus subfamilies*, J. General Virology, 71 (1990), pp. 841–850.

[12] S. Hannenhalli, C. Chappey, E. Koonin, and P. Pevzner, *Genome sequence comparison and scenarios for genome rearrangements: A test case*, Genomics, 30 (1995), pp. 299–311.

[13] S. Hannenhalli and P. Pevzner, *Transforming cabbage into turnip*, in Proc. 27th Annual ACM Symposium on Theory of Computing, ACM, New York, 1995, pp. 178–179.

[14] M. Heydari and I. H. Sudborough, *On sorting by prefix reversals and the diameter of pancake networks*, manuscript, 1993.

[15] M. Jerrum, *The complexity of finding minimum-length generator sequences*, Theoret. Comput. Sci., 36 (1985), pp. 265–289.

[16] S. Karlin, E. S. Mocarski, and G. A. Schachtel, *Molecular evolution of herpesviruses: Genomic and protein sequence comparisons*, J. Virology, 68 (1994), pp. 1886–1902.

[17] J. Kececioglu and D. Sankoff, *Exact and approximation algorithms for the inversion distance between two permutations*, Algorithmica, 13 (1995), pp. 180–210.

[18] J D. Kececioglu and R. Ravi, *Of mice and men: Evolutionary distances between genomes under translocations*, in Proc. Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California, SIAM, Philadelphia, PA, 1995, pp. 604–613.

[19] M. F. Lyon, *X-Chromosome inactivation and the location and expression of X-linked genes*, Amer. J. Hum. Genet., 42 (1988), pp. 8–16.

[20] D. J. McGeoch, *Molecular evolution of large DNA viruses of eukaryotes*, Seminars in Virology, 3 (1992), pp. 399–408.

[21] J. H. Nadeau and B. A. Taylor, *Lengths of chromosomal segments conserved since divergence of man and mouse*, Proc. Nat. Acad. Sci. USA, 81 (1984), pp. 814–818.

[22] S. Ohno, *Sex Chromosomes and Sex-Linked Genes*, Springer-Verlag, Heidelberg, 1967.

[23] J. D. Palmer and L. A. Herbon, *Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence*, J. Molecular Evolution, 27 (1988), pp. 87–97.

[24] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B. F. Lang, and R. Cedergren, *Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome*, Proc. Nat. Acad. Sci. USA, 89 (1992), pp. 6575–6579.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 241–264, May 1998

005

PERFECT FACTORS FROM CYCLIC CODES AND INTERLEAVING*

CHRIS J. MITCHELL† AND KENNETH G. PATERSON‡

Abstract. In this paper, we introduce new construction methods for Perfect Factors. These are based on the theory of cyclic codes, interleaving techniques and the Lempel homomorphism. The constructions enable us to settle the existence question for Perfect Factors for window sizes at most six.

Key words. de Bruijn sequence, de Bruijn graph, window sequence, Perfect Factor, cyclic code, Lempel homomorphism, interleaving

AMS subject classifications. 05C70, 05C38, 94A99, 68R10, 94A55

PII. S089548019630649X

1. Introduction.

In this paper we address the existence question for Perfect Factors. Perfect Factors, i.e., sets of uniformly long cycles whose elements are drawn from an alphabet of size c and in which every possible v-tuple (or "window") of elements occurs exactly once, are of significance for two main reasons (apart from combinatorial interest in their own right).

- They can be used to construct Perfect Maps (or two-dimensional de Bruijn arrays), see, for example, [1, 3, 10, 11], which are of practical importance in certain position-location applications.
- They are special cases of Perfect Maps themselves, and hence their existence is of significance in deciding whether Perfect Maps exist for all parameter sets satisfying certain simple necessary conditions (it has recently been established that these necessary conditions are sufficient for prime power size alphabets, [13, 14]).

It has been conjectured [7] that the simple necessary conditions for the existence of a Perfect Factor (see Lemma 1.3 below) are sufficient for all finite alphabets and for all window sizes. Work towards a proof of this conjecture has progressed along two fronts: first, the conjecture has been shown to be true for specific classes of alphabet size c (for every v), and second, the conjecture has been shown to be true for small values of v regardless of the alphabet size.

The truth of the conjecture was established by Etzion [1] for $c = 2$ and by Paterson [12] in the case where c is a prime power. Further progress was made by Mitchell, who introduced two auxiliary classes of combinatorial objects: Perfect Multifactors (PMFs) [7] and Generalized Perfect Factors (GPFs) [8], which can be combined in various ways to yield Perfect Factors. Powerful constructions for PMFs and GPFs have been given in [7, 8]. An important consequence of this latter work is that the existence question for any particular v can be reduced to an existence question concerning a finite number of "small" parameter sets (see section 7.1 below). In [7, 8] these ideas were used to settle the existence question for $v \leq 4$.

*Received by the editors July 10, 1996; accepted for publication (in revised form) February 20, 1997.

http://www.siam.org/journals/sidma/11-2/30649.html

†Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK (cjm@dcs.rhbnc.ac.uk).

‡Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ, U.K. (kp@hplb.hpl.hp.com). The research of this author was supported by Lloyd's of London Tercentenary Foundation while he was a Research Fellow at the Department of Mathematics, Royal Holloway, University of London.

In this paper, we continue to attack the existence question for Perfect Factors. We introduce three new construction methods for PMFs and GPFs. The first of these uses cyclic codes to construct sequences (section 2), the second is based on interleaving (sections 3 and 4), and the third uses a generalization of the Lempel homomorphism (section 5). We show how these methods can be combined to efficiently analyze the parameter sets required to settle the existence question for $v \leq 6$ (section 7). We also apply our methods to the cases $v = 7$ and $v = 8$, resolving the existence question in all but two cases.

1.1. Notation. We first set up some notation which we will use throughout the paper.

We are concerned here with c-ary periodic sequences, where by c-ary we mean sequences whose elements are drawn from the set $Z_c = \{0, 1, \ldots, c-1\}$. We refer throughout to c-ary cycles of period n, by which we mean periodic sequences $\mathbf{s} = [s_0, s_1, \ldots, s_{n-1}]$ where $s_i \in \{0, 1, \ldots, c-1\}$ for every i, $(0 \leq i < n)$. The least period of such a cycle is defined to be the least positive integer such that $s_i = s_{i+t}$ for all $0 \leq i < n$ (subscripts modulo n).

If $\mathbf{t} = (t_0, t_1, \ldots, t_{v-1})$ is a c-ary v-tuple, i.e., $t_i \in \{0, 1, \ldots, c-1\}$ for every i, $(0 \leq i < v)$, and $\mathbf{s} = [s_0, s_1, \ldots, s_{n-1}]$ is a c-ary cycle of period n $(n \geq v)$, then we say that $\mathbf{t}$ *occurs in* $\mathbf{s}$ *at position* j if and only if

$$t_i = s_{i+j}$$

for every i, $(0 \leq i < v)$, where $i + j$ is computed modulo n.

If $\mathbf{s}$ and $\mathbf{s}'$ are two v-tuples, then we write $\mathbf{s} + \mathbf{s}'$ for the v-tuple obtained by element-wise adding together the two tuples. Similarly, if a is any integer, we write $a\mathbf{s}$ for the tuple obtained by element-wise multiplying the tuple $\mathbf{s}$ by a. Again, if we write $\mathbf{t} = \mathbf{s} \bmod k$, then $\mathbf{t}$ is the tuple obtained by reducing every element in $\mathbf{s}$ modulo k. An exactly analogous interpretation should be used for arithmetic operations on cycles.

If $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{t-1}$ are t cycles all of period n, and if $\mathbf{s}_i = [s_{i0}, s_{i1}, \ldots, s_{i(n-1)}]$ $(0 \leq i < t)$, then $\mathcal{I}(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{t-1})$ denotes the t-fold interleaving of these cycles, i.e., $\mathcal{I}(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{t-1}) = [s_{00}, s_{10}, \ldots, s_{(t-1)0}, s_{01}, s_{11}, \ldots, s_{(t-1)(n-1)}]$, a cycle of period nt.

We define the left shift operator E acting on cycles of period n as follows. The action of E on $\mathbf{s}$, denoted $E\mathbf{s}$, is the cycle whose ith term is s_{i+1} (subscripts being computed modulo n). For $m \geq 2$, we define the action of E^m on $\mathbf{s}$ by writing $E^m\mathbf{s} = E(E^{m-1}\mathbf{s})$. For any polynomial $f(X) = \sum_{i=0}^{m} a_i X^i$ with coefficients in Z_c, we define the action of the operator $f(E)$ on $\mathbf{s}$ to be the cycle $a_0\mathbf{s} + a_1 E\mathbf{s} + \cdots + a_m E^m\mathbf{s}$.

We define a *truncation operator* operating on cycles. Let $\mathbf{s} = [s_0, s_1, \ldots, s_{nt-1}]$ be a cycle, and t be the least positive integer such that $E^t(\mathbf{s}) = \mathbf{s}$, i.e., t is the least period of $\mathbf{s}$. Then let $\mathcal{T}(\mathbf{s}) = [s_0, s_1, \ldots, s_{t-1}]$. Any cycle $\mathbf{s}$ of period n and least period t is equally well represented by the cycle $\mathcal{T}(\mathbf{s})$.

The weight of a period n cycle is defined to be the sum of its n elements evaluated in Z_c. Notice that

$$\frac{E^n - 1}{E - 1}\mathbf{s} = (E^{n-1} + \cdots + E + 1)s = \left[\sum_{i=0}^{n-1} s_i, \sum_{i=0}^{n-1} s_i, \ldots, \sum_{i=0}^{n-1} s_i\right]$$

so that $\frac{E^n-1}{E-1}\mathbf{s}$ is a constant cycle whose terms equal the weight of $\mathbf{s}$. We say that a set of period n cycles over Z_c is a *constant weight set* if each of the cycles in the

set has the same weight. We define the *total weight* of the set to be the sum of the weights of the cycles in the set.

In addition, we use the notation (m, n) to represent the *Greatest Common Divisor* of m and n (given that m, n are a pair of positive integers or a pair of polynomials over some field). By convention, $(0, n) = n$.

1.2. Fundamental definitions and results.

1.2.1. De Bruijn sequences. We first have the following.

DEFINITION 1.1. *A c-ary* de Bruijn sequence of span v *is a c-ary cycle of period c^v which contains c^v distinct v-tuples in a period of the cycle; equivalently, every possible c-ary v-tuple occurs precisely once in a period of a de Bruijn sequence.*

It has long been known that c-ary span v de Bruijn sequences exist for all values of $c > 1$ and $v > 0$ (see [2] for a proof of this result and a comprehensive survey of the long and interesting history of de Bruijn sequences).

1.2.2. Perfect Factors. We next define a generalization of de Bruijn sequences, the construction of which is the main theme of this paper.

DEFINITION 1.2. *Suppose n, c, and v are positive integers (where we also assume that $c \geq 2$). An (n, c, v)–Perfect Factor, or simply an (n, c, v)–PF, is a collection of c^v/n c-ary cycles of period n with the property that every c-ary v-tuple occurs in one of these cycles.*

Note that, because a Perfect Factor contains exactly c^v/n cycles, and because there are clearly c^v different c-ary v-tuples, each v-tuple will actually occur exactly once somewhere in the collection of cycles (and hence all the cycles are distinct). Also observe that a (c^v, c, v)–PF is simply a c-ary span v de Bruijn sequence.

The following necessary conditions for the existence of a Perfect Factor are trivial to establish.

LEMMA 1.3. *Suppose A is a (n, c, v)–PF. Then*

1. *$n|c^v$, and*
2. *$v < n$ (or $n = v = 1$).*

CONJECTURE 1.4 (see [7, Conjecture 1.4]). *The conditions of Lemma* 1.3 *are sufficient for the existence of an (n, c, v)–PF.*

We next give a simple but useful construction for Perfect Factors.

CONSTRUCTION 1.5. *Suppose n and c are integers greater than* 1*, where $n|c^{n-1}$. Let A^* be the set of all c-ary cycles of period n with the property that the sum of the elements in each cycle is congruent to* 1 *modulo c. If $\mathbf{a}, \mathbf{a}' \in A^*$, then define $\mathbf{a} \sim \mathbf{a}'$ if and only if $\mathbf{a} = E^s\mathbf{a}'$ for some s. It is simple to see that $\sim$ is an equivalence relation on the elements of A^*, and hence define A to be a set of $\sim$-representatives from A^*.*

LEMMA 1.6. *If n, c and A are as in Construction* 1.5*, then A is an $(n, c, n-1)$–PF.*

Proof. Consider any c-ary $(n-1)$-tuple. It clearly occurs at position 0 in a unique cycle in A^*, and can only occur once in any cycle of A^*. Hence, it occurs once within a unique cycle in A, and the result follows. □

COROLLARY 1.7. *The conditions of Lemma* 1.3 *are sufficient for the existence of an (n, c, v)–PF when $v = n - 1$.*

In view of the first condition in Lemma 1.3, we can assume that the prime factorizations of c and n are

$$c = \prod_{i=1}^{t} {p_i}^{r_i} \quad \text{and} \quad n = \prod_{i=1}^{t} {p_i}^{s_i},$$

where $0 \leq s_i \leq r_i v$ for each i.

We discuss next the extent to which Conjecture 1.4 is known to be true. The case where $v = 1$ is clearly trivial, and we have dealt with the case $v = n - 1$ in Corollary 1.7. The conditions of Lemma 1.3 are known to be sufficient when $c = 2$ [1] and when c is a power of a prime [12]. It was also proved in [12] that the conditions of Lemma 1.3 are sufficient when $p_i^{s_i} > v$ for *every* index i. In [7] this result has been improved to establish the sufficiency of the conditions of Lemma 1.3 whenever $p_i^{s_i} > v$ for *at least one* index i.

THEOREM 1.8 (see Theorem 7.1 of [7]). *An (n, c, v)–PF can be constructed for any n, c, and v satisfying $v < n|c^v$ and $c > 1$, as long as $v < p^s$ and $p^s|n$ for some prime p and some positive integer s.*

This immediately implies that Conjecture 1.4 holds for $v = 2$ and that the conjecture remains open only for periods $n = \prod_{i=1}^{t} {p_i}^{s_i}$ for which $p_i^{s_i} \leq v$ for each $1 \leq i \leq t$.

The truth of Conjecture 1.4 has also been established for every c when $v \leq 4$ [8]. Certain other cases for $v = 6$ and larger composite v have recently been dealt with in [9].

1.2.3. Perfect Multifactors. We define a related set of combinatorial objects, first introduced in [7].

DEFINITION 1.9. *Suppose m, n, c and v are positive integers satisfying $m|c^v$ and $c \geq 2$. An (m, n, c, v)–Perfect Multifactor, or simply an (m, n, c, v)–PMF, is a collection of c^v/m c-ary cycles of period mn with the property that for every c-ary v-tuple $\mathbf{t}$, and for every integer j in the range $0 \leq j < n$, $\mathbf{t}$, occurs at a position $p \equiv j \pmod{n}$ in one of these cycles.*

Note that, because a PMF contains c^v/m cycles (each of period mn and hence "containing" mn v-tuples), and because there are clearly c^v different c-ary v-tuples, each v-tuple will actually occur exactly n times in the collection of cycles, once in each of the possible position congruency classes (mod n). This also implies that all the cycles are distinct.

REMARK 1.10. *It should be clear that an $(m, 1, c, v)$–PMF is precisely equivalent to an (m, c, v)–PF. In addition, observe that a $(1, n, c, v)$–PMF is simply a collection of c^v c-ary cycles of period n with the property that every c-ary v-tuple occurs at every possible position in one of the cycles.*

The following necessary conditions for the existence of a Perfect Multifactor are trivial to establish.

LEMMA 1.11 (see [7]). *Suppose A is an (m, n, c, v)–PMF. Then*

(i) *$m|c^v$, and*

(ii) *$v < mn$ (or $m = 1$ and $v = n$).*

It has been conjectured in [7] that the above necessary conditions are sufficient for the existence of a PMF. The following result establishes the existence conjecture whenever $n \geq v$ (and in particular for the special case $m = 1$).

THEOREM 1.12 [7]. *Suppose n, c, v are positive integers ($c \geq 2$ and $n \geq v$). Then there exists an (m, n, c, v)–PMF for every positive integer m satisfying $m|c^v$.*

We next show how an established construction technique can be used to produce Perfect Multifactors. A slightly different formulation of the following method was previously given as Construction E in [8].

CONSTRUCTION 1.13. *Suppose c, d, σ, τ, and μ are positive integers where $c \geq 2$ and $d \geq 2$, and let*

$$A = \{\mathbf{a}_i : 0 \leq i < \sigma\}$$

be a set of σ c-ary cycles of period μ, and

$$B = \{\mathbf{b}_i : 0 \leq i < \tau\}$$

be a set of τ d-ary cycles also of period μ. Now let

$$C = \{\mathbf{s}_{ij} : 0 \le i < \sigma,\ 0 \le j < \tau\}$$

be the set of cd-ary cycles of period μ defined by

$$\mathbf{s}_{ij} = \mathbf{a}_i + c\mathbf{b}_j.$$

THEOREM 1.14. *Suppose c, d, σ, τ, μ, A, and B satisfy the conditions of Construction* 1.13. *Suppose also that, for some $v \ge 1$, A is an (m, n, c, v)–PMF and B is a $(1, mn, d, v)$–PMF. If C is derived from A and B (with $\sigma = c^v/m$, $\tau = d^v$ and $\mu = mn$) using Construction* 1.13, *then C is an (m, n, cd, v)–PMF.*

Proof. Suppose $\mathbf{t}$ is a (cd)-ary v-tuple. Let $\mathbf{u} = \mathbf{t} \bmod c$, and let $\mathbf{w} = (\mathbf{t} - \mathbf{u})/c$. Then $\mathbf{u}$ is a c-ary v-tuple and $\mathbf{w}$ is a d-ary v-tuple and we have

$$\mathbf{t} = \mathbf{u} + c\mathbf{w}.$$

Now suppose $0 \le i < n$; then we need to show that $\mathbf{t}$ occurs at a position congruent to i modulo n in some cycle of C. Now, since A is an (m, n, c, v)–PMF, $\mathbf{u}$ occurs at a position congruent to i modulo n in some cycle of A; say $\mathbf{s}$ occurs at position $i + \ell n$ in cycle $\mathbf{a}_j$ for some ℓ and j. In addition, since B is a $(1, mn, d, v)$–PMF, $\mathbf{t}$ occurs at position $i + \ell n$ in some cycle, say $\mathbf{b}_k$, of A'. It is then immediate to see that $\mathbf{t}$ occurs at position $i + \ell n$ in $\mathbf{s}_{jk}$, and the result follows. □

Next observe that, by Theorem 1.12, a $(1, mn, d, v)$–PMF exists whenever $mn \ge v$, and hence by combining Theorem 1.14 with Theorem 6.5 of [7], we have the following.

THEOREM 1.15. *Suppose there exists an (m, n, c, v)–PMF. Then, for every $\beta \ge 1$ and every $d \ge 1$, there exists an $(m, \beta n, cd, v)$–PMF, given that $(\beta, m) = 1$.*

1.2.4. Generalized Perfect Factors. We now define yet another class of combinatorial objects, the definition of which is a generalization of the notion of Perfect Factor (as is the definition of PMF). We subsequently use these objects to help construct new Perfect Factors.

DEFINITION 1.16. *Suppose m, n, c, and v are positive integers satisfying $m|c^v$ and $c \ge 2$. An (m, n, c, v)–Generalized Perfect Factor, or simply an (m, n, c, v)–GPF, is a collection of c^v/m c-ary cycles of period mn with the following property. For every c-ary v-tuple $\mathbf{t}$, there exists an integer j in the range $0 \le j < m$ such that for every i $(0 \le i < n)$ $\mathbf{t}$ occurs at position $j + im$ in one of these cycles.*

Note that, because a GPF contains exactly c^v/m cycles (each "containing" mn v-tuples), and because there are clearly c^v different c-ary v-tuples, each v-tuple will actually occur exactly n times in the set of cycles, once in each position $j + im$ $(0 \le i < n)$. This immediately implies that all the cycles are distinct.

REMARK 1.17. *It should be clear that*

(i) *an $(m, 1, c, v)$–GPF is precisely equivalent to an (m, c, v)–PF, and*

(ii) *a $(1, n, c, v)$–GPF is precisely equivalent to a $(1, n, c, v)$–PMF.*

The following result is also straightforward to prove.

THEOREM 1.18 (see [8]). *Suppose A is an (m, n, c, v)–GPF, where $(m, n) = 1$. Then A is also an (m, n, c, v)–PMF.*

The following necessary conditions for the existence of a GPF are trivial to establish.

LEMMA 1.19 (see [8]). *Suppose A is an (m, n, c, v)–GPF. Then*

(i) *$m|c^v$, and*

(ii) *$v < mn$ (or $m = 1$ and $v = n$).*

It is tempting at this point to conjecture that the necessary conditions specified in Lemma 1.19 for the existence of an (m, n, c, v)–GPF are sufficient. However, as established in [8], this is not true. Nevertheless we do have the following (constructive) existence results for GPFs.

THEOREM 1.20 (see [8, Theorems 19 and 21]). *Suppose there exists an (m, n, c, v)–GPF. Then, for every $\lambda \geq 1$ and every $d \geq 1$, there exists an $(m, \lambda n, cd, v)$–GPF.*

This result provides a useful analogue to Theorem 1.15.

We also have the following result, which we use repeatedly below.

THEOREM 1.21 (see Theorem 16 of [8]). *Suppose m, n, c, and v are positive integers satisfying $m|c^v$ and $c \geq 2$, and*

$$A = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1}\}$$

is a set of c-ary cycles of least periods $\ell_0, \ell_1, \ldots, \ell_{t-1}$, respectively, with the property that $m|\ell_i|mn$ for every i, $0 \leq i < t$, and with the property that every c-ary v-tuple occurs precisely once in the set of cycles. Then, for every i $(0 \leq i < t)$ let $\mathbf{w}_i$ be defined as $\mathbf{a}_i$ concatenated with itself mn/ℓ_i times. Next, let

$$\mathbf{b}_{ij} = \mathbf{E}^{jm}(\mathbf{w}_i)$$

for every j, $(0 \leq j < \ell_i/m)$. Finally, let

$$B = \{\mathbf{b}_{ij} \ : \ 0 \leq i < t,\ 0 \leq j < \ell_i/m\}.$$

Then B is an (m, n, c, v)–GPF.

We next observe that Construction 1.13 can also be used to produce new GPFs.

THEOREM 1.22. *Suppose c, d, σ, τ, μ, A, and B satisfy the conditions of Construction* 1.13. *Suppose also that, for some $v \geq 1$, A is an (m_1, n_1, c, v)–GPF and B is an (m_2, n_2, d, v)–GPF, where $m_1 n_1 = m_2 n_2$ and $(m_1, m_2) = 1$ (and hence $m_2|n_1$). If C is derived from A and B (with $\sigma = c^v/m_1$, $\tau = d^v/m_2$ and $\mu = m_1 n_1 = m_2 n_2$) using Construction* 1.13, *then C is an $(m_1 m_2, n_1/m_2, cd, v)$–GPF.*

Proof. Suppose $\mathbf{t}$ is a (cd)-ary v-tuple. We need to exhibit an integer j with $0 \leq j < m_1 m_2$ such that for every i $(0 \leq i < n_1/m_2)$, $\mathbf{t}$ occurs at position $j + im_1 m_2$ in one of the cycles $\mathbf{s}_{ij}$ of C. Let $\mathbf{u} = \mathbf{t} \bmod c$, and let $\mathbf{w} = (\mathbf{t} - \mathbf{u})/c$. Then $\mathbf{u}$ is a c-ary v-tuple and $\mathbf{w}$ is a d-ary v-tuple, and we have

$$\mathbf{t} = \mathbf{u} + c\mathbf{w}.$$

Now there exists an integer j_1 with $0 \leq j_1 < m_1$ such that for every i $(0 \leq i < n_1)$, $\mathbf{u}$ occurs at position $j_1 + im_1$ in a cycle of A. There also exists an integer j_2 with $0 \leq j_2 < m_2$ such that for every i $(0 \leq i < n_2)$, $\mathbf{w}$ occurs at position $j_2 + im_2$ in a cycle of B. Since $(m_1, m_2) = 1$, by the Chinese Remainder Theorem there is a unique j with $0 \leq j < m_1 m_2$ that satisfies the pair of congruences:

$$\begin{aligned} j &\equiv j_1 \pmod{m_1} \\ j &\equiv j_2 \pmod{m_2}. \end{aligned}$$

Suppose i with $0 \leq i < n_1/m_2$ is fixed. It is certainly true that there is a cycle $\mathbf{a}_i \in A$ and a cycle $\mathbf{b}_j \in B$ such that $\mathbf{u}$ appears in $\mathbf{a}_i$ and $\mathbf{w}$ appears in $\mathbf{b}_j$ at position $j + im_1 m_2$. The v-tuple $\mathbf{t}$ then appears at position $j + im_1 m_2$ in the cycle $\mathbf{s}_{ij}$. It follows that the set of cycles C forms an $(m_1 m_2, n_1/m_2, cd, v)$–GPF. □

REMARK 1.23. *Observe that, in the case $m_2 = n_1$ (and hence $m_1 = n_2$ and the constructed GPF is actually a PF), by Theorem* 1.18 *the above result coincides with Theorem* 23 *of* [8].

1.3. Using PMFs and GPFs to construct Perfect Factors. We conclude these introductory remarks by showing how PMFs and GPFs can be used to construct Perfect Factors. We start with an existence result for Perfect Factors from [8, Theorem 23]. This result, which derives from a simple application of Construction 1.13, is central to the work in this paper.

THEOREM 1.24. *Suppose there exists an (ν, μ, c, v)–GPF and a (μ, ν, d, v)–PMF. Then there exists a $(\mu\nu, cd, v)$–PF.*

Now, from Remark 1.17(i) and Theorem 1.20, it should be clear that if there exists an (n, c, v)–PF, then we can construct an (n, m, c, v)–GPF for every positive integer m. Combining this observation with Theorem 1.24 we obtain as an immediate corollary the following result, first given as Theorem 5.2 of [7].

THEOREM 1.25. *If there exists an (n, c, v)–PF and an (m, n, d, v)–PMF, then there exists an (mn, cd, v)–PF.*

Since, by Theorem 1.12, a $(1, n, d, v)$–PMF exists for every n, d, and v ($n \geq v$ and $d > 1$), we immediately have the following.

COROLLARY 1.26. *If there exists an (n, c, v)–PF, then there exists an (n, cd, v)–PF for every $d \geq 1$.*

2. Sequence sets from cyclic codes. In this section, we will give constructions for GPFs and PMFs that are based on the theory of cyclic codes. We refer to [6, Chapter 7] for the necessary background information that we assume here.

Throughout, we assume that n is an integer and p is a prime with $n = p^l s$, $(p, s) = 1$. We work with p-ary cycles and codes of length n. We define a cyclic code of length n over Z_p to be an ideal C in the ring $Z_p[X]/(X^n - 1)$. This ring is a principal ideal domain and so C has a generator g. We can associate with g a polynomial $g(X) \in Z_p[X]$ with $\deg g(X) \leq n$ and $g(X) | X^n - 1$. Let $k = \deg g(X)$. We can then write

$$C = \{c(X)g(X) \bmod X^n - 1 : c(X) \in Z_p[X], \deg c(X) < n - k\},$$

and regard C as a set of polynomials of degree at most $n - 1$. The code C is a linear code with dimension $n - k$. We can associate with each polynomial $a(X) = a_0 + a_1 X + \cdots + a_{n-1}X^{n-1}$ the p-ary n-tuple $\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$. We call the set of tuples obtained from the elements of C in this way the codewords of C. We can regard the codewords as a set of cycles. Then it is easy to see that the action of E on a cycle is equivalent to that of multiplication of the corresponding polynomial by $X^{n-1} \bmod X^n - 1$. Notice also that the weight of a cycle $\mathbf{a}$ is equal to $a(1)$, the value of $a(X)$ evaluated at 1.

We need to examine the tuples appearing in the cycles obtained from C. Because of the linearity of C, there is a $(n - k) \times n$ matrix G (called the generator matrix of C) such that every codeword of C is a linear combination of the rows of G. We can assume that G is of the form $[I_{n-k}|A]$ where I_{n-k} denotes the $(n-k)\times(n-k)$ identity matrix and A is an $(n-k) \times k$ matrix. Thus if the $n-k$ values $a_0, a_1, \ldots, a_{n-k-1}$ are specified, then there is a unique n-tuple $\mathbf{a} = [a_0, a_1, \ldots, a_{n-k-1}, a_{n-k}, \ldots, a_{n-1}]$ such that $\mathbf{a} \in C$. This shows that every p-ary $(n-k)$-tuple occurs exactly once in position zero of a codeword of C. Since the set C is closed under cyclic shifting, the same is true of any position i with $0 \leq i < n$. This immediately shows that the set of cycles obtained from any cyclic code C form a $(1, n, p, n - k)$–PMF.

2.1. Some preliminaries. We will use cosets of cyclic codes to obtain GPFs and PMFs. The coset of C defined by polynomial $b(X)$ is defined to be the set $C + b(X)$ (addition modulo $X^n - 1$). We have the following lemmas.

LEMMA 2.1. *Let C be a length n cyclic code with generator polynomial $g(X)$. Then the coset $C + b(X)$ is closed under cyclic shifting by all multiples of t positions if and only if $a(X)g(X) \equiv b(X)(X^t - 1) \pmod{X^n - 1}$ for some $a(X)$.*

Proof. The elements of $C + b(X)$ are the polynomials $c(x)g(X) + b(X)$, where $\deg c(X) < n - k$, and a set S of polynomials is closed under cyclic shifting by all multiples of t positions if and only if $X^t S \equiv S \pmod{X^n - 1}$. Now $X^t(c(X)g(X) + b(X)) = X^t c(X)g(X) + X^t b(X)$ lies in $C + b(X)$ if and only if $X^t b(X) \equiv a(X)g(X) + b(X) \pmod{X^n - 1}$ for some polynomial $a(X)$, which in turn is equivalent to writing $b(X)(X^t - 1) \equiv a(X)g(X) \pmod{X^n - 1}$. □

LEMMA 2.2. *Suppose that $r \leq l$, and that for every t with $t|n$ and $(t, p^l)|p^{r-1}$, we have that the polynomial*

$$\left(g(X), \frac{X^n - 1}{X^t - 1}\right)$$

does not divide $b(X)$. Then every cycle derived from the coset $C + b(X)$ has least period divisible by p^r.

Proof. Suppose that the condition in the statement of the lemma holds. Then for any t with $t|n$ and $(t, p^l)|p^{r-1}$, we have that $(g(X), \frac{X^n-1}{X^t-1})$ does not divide the polynomial $c(X)g(X) + b(X)$ for any $c(X)$. Hence, for every $s(X) \in C + b(X)$, $((X^t - 1)g(X), X^n - 1)$ does not divide $s(X)(X^t - 1)$. Hence,

$$s(X)(X^t - 1) \neq 0 \bmod X^n - 1, \quad \text{for every } s(X) \in C + b(X).$$

It follows from this that no cycle from $C + b(X)$ has least period divisible by t. Since every such cycle has least period dividing $n = p^l s$, we deduce that p^r must divide the period of every cycle from $C + b(X)$. □

LEMMA 2.3. *Suppose that, for every t with $t|n$ $(t \neq n)$, the polynomial*

$$\left(g(X), \frac{X^n - 1}{X^t - 1}\right)$$

does not divide $b(X)$. Then every cycle derived from the coset $C + b(X)$ has least period n.

Proof. Using exactly the same argument as in the proof of Lemma 2.2, no cycle from $C + b(X)$ has least period divisible by t for any $t|n$ $(t \neq n)$. But every cycle has least period dividing n and the lemma follows. □

2.2. A cyclic code construction for GPFs. We now have a construction for GPFs.

CONSTRUCTION 2.4. *Let n be an integer and p a prime with $n = p^l s$, $(p, s) = 1$. Suppose $1 \leq r \leq l$. Let $g(X)$ be a polynomial of degree k in $Z_p[X]$ with $g(X)|X^n - 1$ and suppose $X - 1$ divides $g(X)$ exactly λ times, where $1 \leq \lambda \leq p^l - p^{r-1}$. Let C denote the length n p-ary code with generator polynomial $g(X)$. Let $b(X) = g(X)/(X - 1)$ and define $S = C + b(X)$. We regard S as a set of p-ary cycles of period n. Define an equivalence relation $\sim$ on S by writing $\mathbf{x} \sim \mathbf{y}$ if and only if $\mathbf{x} = E^t(\mathbf{y})$ for some t. Let R be a set of $\sim$-class representatives. Finally, let $A = \{\mathcal{T}(\mathbf{a}) : \mathbf{a} \in R\}$.*

THEOREM 2.5. *Let A be constructed as in Construction 2.4. Then A is a collection of cycles such that*

- *every p-ary $(n - k)$-tuple occurs exactly once in a cycle of A,*
- *every cycle of A has a least period t satisfying $p^r|t|n$.*

The result of applying Theorem 1.21 *to* A *is a* $(p^r, n/p^r, p, n-k)$*–GPF in which each cycle has weight equal to* $b(1)$*.*

Proof. Define $g(X)$ and the sets S and A as in Construction 2.4. Notice that each cycle in S has weight equal to $c(1)g(1) + b(1)$ for some polynomial $c(X)$. But $g(1) = 0$ (because $X - 1|g(X)$), so S has constant weight equal to $b(1)$.

Suppose $t|n$ and $(t, p^l)|p^{r-1}$. Then in $Z_p[X]$, $X^t - 1$ is divisible by $X - 1$ at most p^{r-1} times, while $X^n - 1$ is divisible by $X - 1$ exactly p^l times. It follows that the polynomial $(g(X), \frac{X^n-1}{X^t-1})$ is divisible by $X-1$ exactly λ times. But $b(X) = g(X)/X-1$ is divisible by $X-1$ exactly $\lambda-1$ times, so $(g(X), \frac{X^n-1}{X^t-1})$ does not divide $b(X)$. From Lemma 2.2, each cycle in S has least period divisible by p^r. Therefore the cycles in A all have periods that are divisible by p^r.

We also know that every $(n-k)$-tuple appears exactly once in position 0 of some cycle derived from the code C, and so the same is true of $S = C + b(X)$. Moreover, because of the choice for $b(X)$, by Lemma 2.1 S is closed under cyclic shifting. It follows that the $(n-k)$-tuples occurring in a cycle $\mathbf{a}$ of R are exactly the $(n-k)$-tuples that occur in position 0 of the cycles in the $\sim$-class containing $\mathbf{a}$. Thus the set A, derived from R by truncation, has the property that every p-ary $(n-k)$-tuple occurs exactly once as a subsequence of a cycle in A.

Theorem 1.21 guarantees that A can be used to produce a $(p^r, n/p^r, p, n-k)$–GPF. Each cycle of this GPF is obtained from a cycle of A by concatenation and shifting, and so in fact is a cycle of S. Since the cycles of S all have weight $b(1)$, so do the cycles of the GPF. □

The parameters of the GPFs that can be obtained from Construction 2.4 depend heavily on the degrees of the factors of $X^n - 1$ in $Z_p[X]$ (since we require a degree k polynomial $g(X)$ with $X - 1|g(X)|X^n - 1$). The complete factorization of $X^n - 1$ in $Z_p[X]$ is known [5, Theorems 2.45 and 2.47]: if $n = p^l s$ with $(s, p) = 1$, then $X^n - 1 = (X^s - 1)^{p^l}$ and

$$X^s - 1 = \prod_{d|s} C_d(X),$$

where $C_d(X)$ of degree $\phi(d)$ is the *dth cyclotomic polynomial over* $Z_p[X]$. The polynomial $C_d(X)$ has $\phi(d)/e$ irreducible factors of degree e, where e is the least positive integer such that $p^e \equiv 1 \bmod d$.

EXAMPLE 2.6. *We aim to construct a* $(2, 3, 2, 3)$*–GPF and a* $(3, 2, 3, 3)$*–GPF. By Theorem* 1.22*, if these are combined using Construction* 1.13*, then we obtain a* $(6, 6, 3)$*–PF.*

We take $n = 6$*,* $p = 2$ *and find that* $X^6 - 1 = (X + 1)^2(X^2 + X + 1)^2$ *in* $Z_2[X]$*. We take* $r = 1$ *and* $g(X) = (X + 1)(X^2 + X + 1)$ *in Construction* 2.4 *to obtain a* $(2, 3, 2, 3)$*–GPF in which each cycle has weight* 1*.*

Similarly, $X^6 - 1 = (X - 1)^3(X + 1)^3$ *in* $Z_3[X]$*. We take* $r = 1$ *and* $g(X) = (X - 1)(X + 1)^2$ *in Construction* 2.4 *to obtain a* $(3, 2, 3, 3)$*–GPF in which each cycle has weight* 1*.*

Combining these two GPFs using Construction 1.13*, we obtain a* $(6, 6, 3)$*–PF.*

We now have the following theorem, whose proof gives a constructive method for obtaining Perfect Factors having prime window size v. This theorem will be useful when we come to analyze parameter sets for small v in section 7.

THEOREM 2.7. *Suppose that* p *is a prime with* $p|c$ *and* p *divides* n *exactly once. Suppose further that the parameters* (n, c, p) *satisfy the necessary conditions of Lemma* 1.3*. Finally suppose that for some prime* q *with* $q|(n/p)$*, we have* $p \equiv 1 \pmod q$*. Then there exists an* (n, c, p)*–PF.*

Proof. Let n, c and p be as above. We aim to use Construction 2.4 to obtain a $(p, n/p, p, p)$–GPF.

Consider the factorization of $X^n - 1$ in $Z_p[X]$. Because q satisfies $p \equiv 1 \pmod{q}$, the q-th cyclotomic polynomial $C_q(X)$ over $Z_p[X]$ has $q - 1 \geq 1$ linear factors. Let $X - \alpha$, $\alpha \neq 1$, be one of these. Since $q \geq 2$ divides n/p, $C_1(X)C_q(X)$ divides $X^{n/p} - 1$. We deduce that $X^n - 1 = (X-1)^p(X-\alpha)^p h(X)$ for some polynomial $h(X)$ where $(X - 1, h(X)) = 1$. We take

$$g(X) = \frac{(X^n - 1)}{(X-1)^{p-1}(X-\alpha)}$$

so that $X - 1$ divides $g(X)$ exactly once and $g(X)$ has degree equal to $n - p$. Taking $\ell = r = 1$ in Construction 2.4, we can obtain a GPF with parameters $(p, n/p, p, p)$.

Now because the parameters (n, c, p) satisfy the necessary conditions of Lemma 1.3 and p divides n exactly once, we have $(n/p)|(c/p)^p$. We can use Theorem 1.12 to deduce that there exists an $(n/p, p, c/p, p)$–PMF. Combining this PMF and the GPF constructed above using Theorem 1.24, we obtain an (n, c, v)–PF. □

2.3. A cyclic code construction for PMFs. We now have a corresponding code construction for PMFs.

CONSTRUCTION 2.8. *Let n, p, and r be nonnegative integers where p is prime, $n > 0$ and $p^r | n$. Let $b(X), g(X) \in Z_p[X]$ ($g(X) | X^n - 1$ and $g(X)$ of degree k), and suppose*

(i) *$g(X) | b(X)(X^{n/p^r} - 1)$, and*

(ii) *$(g(X), \frac{X^n - 1}{X^t - 1})$ does not divide $b(X)$ for any $t | n$ ($t \neq n$).*

Let C denote the length n p-ary code with generator polynomial $g(X)$, and define $S = C + b(X)$. We regard S as a set of p-ary cycles of period n. Finally, define an equivalence relation $\sim$ on S by writing $\mathbf{x} \sim \mathbf{y}$ if and only if $\mathbf{x} = E^{un/p^r}(\mathbf{y})$ for some integer u, and let A be a set of $\sim$-class representatives.

THEOREM 2.9. *Let A be constructed as in Construction 2.8. Then A is a $(p^r, n/p^r, p, n - k)$–PMF.*

Proof. Define $g(X)$ and the sets S and A as in Construction 2.8.

By Lemma 2.3, condition (ii) of the construction immediately implies that each cycle in A has least period n.

We also know that, for every i ($1 \leq i < n$), every $(n - k)$-tuple appears exactly once in position i of some cycle derived from the code C, and so the same is true of $S = C + b(X)$. Moreover, because of the choice for $b(X)$, Lemma 2.1 implies that S is closed under cyclic shifting by multiples of n/p^r positions. Thus, for every i ($1 \leq i < n/p^r$) every $(n - k)$-tuple appears exactly once at a position congruent to i modulo n/p^r in some cycle from the set A.

The result now follows. □

As previously, the parameters of the PMFs that Construction 2.8 allows us to obtain depend heavily on the degrees of the factors of $X^n - 1$ in $Z_p[X]$ (since we require a degree k polynomial $g(X)$ with $g(X) | X^n - 1$).

EXAMPLE 2.10. *We aim to construct a $(2, 3, 2, 4)$–PMF and a $(3, 2, 3, 4)$–GPF. By Theorem 1.24, these can be combined to obtain a $(6, 6, 4)$–PF.*

Using Construction 2.8, we take $r = 1$, $n = 6$, and $p = 2$, and find that $X^6 - 1 = (X+1)^2(X^2 + X + 1)^2$ in $Z_2[X]$. We take $b(X) = 1$ and $g(X) = (X^2 + X + 1)$ to obtain a $(2, 3, 2, 4)$–PMF.

Now, $X^6 - 1 = (X-1)^3(X+1)^3$ in $Z_3[X]$. We take $r = 1$ and $g(X) = (X - 1)(X + 1)$ in Construction 2.4 to obtain a $(3, 2, 3, 4)$–GPF.

Applying Theorem 1.24*, we obtain a* $(6, 6, 4)$*–PF.*

3. An interleaving construction for Perfect Multifactors. We now describe a method for constructing Perfect Multifactors by interleaving the cycles of a (smaller) Perfect Factor. We subsequently use this construction method to help construct Perfect Factors with "new" parameters.

3.1. The construction method.

CONSTRUCTION 3.1. *Suppose* c, n, t*, and* v *are positive integers where* $c \geq 2$ *and* $t \geq 2$*, and let* $A = \{\mathbf{a}_i : 0 \leq i < c^v/n\}$ *be an* (n, c, v)*–PF.*

Now define a set B *containing* c^{tv}/n c*-ary cycles of period* nt *by*

$$B = \{\mathbf{b_{ij}} : \mathbf{i} = (i_0, i_1, \ldots, i_{t-1}),\ (0 \leq i_s < c^v/n);\ \mathbf{j} = (j_0, j_1, \ldots, j_{t-2}),\ (0 \leq j_s < n)\}$$

where

$$\mathbf{b_{ij}} = \mathcal{I}(\mathbf{a}_{i_0}, E^{j_0}\mathbf{a}_{i_1}, \ldots, E^{j_{t-2}}\mathbf{a}_{i_{t-1}}).$$

We then have the following result.

THEOREM 3.2. *Suppose* c, n, t, v*, and* A *satisfy the conditions of Construction* 3.1*. If* B *is constructed from* A *using Construction* 3.1*, then* B *is an* (n, t, c, tv)*–PMF.*

Proof. Suppose $\mathbf{y}$ is any c-ary tv-tuple, and choose any r with $0 \leq r < t$. We need to show that $\mathbf{y}$ occurs at a position congruent to r modulo t in a cycle of B.

First, let

$$\mathbf{y} = \mathcal{I}(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{t-1}),$$

where $\mathbf{x}_u$ is a c-ary v-tuple for every u. Now, $\mathbf{y}$ occurs at position $r + st$ in $\mathbf{b_{ij}}$ (for some $s, \mathbf{i}$ and $\mathbf{j}$) if and only if

$$\mathbf{x}_u \text{ occurs at position } \begin{cases} s \\ s+1 \\ s+1 \end{cases} \text{ in } \begin{cases} E^{j_{u+r-1}}\mathbf{a}_{i_{u+r}} & \text{if } 0 \leq u < t-r, \\ \mathbf{a}_{i_0} & \text{if } u = t-r, \\ E^{j_{u+r-1-t}}\mathbf{a}_{i_{u+r-t}} & \text{if } t-r+1 \leq u < t. \end{cases}$$

Now, since A is an (n, c, v)–PF, there exists a unique pair of values (s, i_0) for which $\mathbf{x}_{t-r}$ occurs at position $s+1$ in $\mathbf{a}_{i_0}$. Given this value of s, then there exist unique pairs of values: (j_{u+r-1}, i_{u+r}) for which

$$\mathbf{x}_u \text{ occurs at position } s \text{ in } E^{j_{u+r-1}}\mathbf{a}_{i_{u+r}}, \qquad (0 \leq u < t-r),$$

and also there exist unique pairs of values: $(j_{u+r-1-t}, i_{u+r-t})$ for which

$$\mathbf{x}_u \text{ occurs at position } s+1 \text{ in } E^{j_{u+r-1-t}}\mathbf{a}_{i_{u+r-t}}, \qquad (t-r+1 \leq u < t).$$

Thus $\mathbf{y}$ occurs at a position congruent to r modulo t in a unique cycle of B, and hence B is an (n, t, c, tv)–PMF. □

EXAMPLE 3.3. *Let* A *be the following set of five* 5*-ary cycles of period* 5*, which constitute a* $(5, 5, 2)$*–PF.*

$$\mathbf{a}_0 = [0\,0\,1\,3\,1],\ \mathbf{a}_1 = [1\,1\,2\,4\,2],\ \mathbf{a}_2 = [2\,2\,3\,0\,3],\ \mathbf{a}_3 = [3\,3\,4\,1\,4],\ \mathbf{a}_4 = [4\,4\,0\,2\,0].$$

Then, by applying Construction 3.1 *with* $t = 2$ *we obtain the following* $(5, 2, 5, 4)$*–PMF (a set of* 125 *cycles of period* 10 *in which every* 5*-ary* 4*-tuple occurs at positions congruent to* 0 *and* 1 *modulo* 2*).*

$\mathbf{b}_{(00)(0)} = [0\,0\,0\,0\,1\,1\,3\,3\,1\,1]$, $\mathbf{b}_{(00)(1)} = [0\,0\,0\,1\,1\,3\,3\,1\,1\,0]$, $\mathbf{b}_{(00)(2)} = [0\,1\,0\,3\,1\,1\,3\,0\,1\,0]$,
$\mathbf{b}_{(00)(3)} = [0\,3\,0\,1\,1\,0\,3\,0\,1\,1]$, $\mathbf{b}_{(00)(4)} = [0\,1\,0\,0\,1\,0\,3\,1\,1\,3]$,
$\mathbf{b}_{(01)(0)} = [0\,1\,0\,1\,1\,2\,3\,4\,1\,2]$, $\mathbf{b}_{(01)(1)} = [0\,1\,0\,2\,1\,4\,3\,2\,1\,1]$, $\mathbf{b}_{(01)(2)} = [0\,2\,0\,4\,1\,2\,3\,1\,1\,1]$,
$\mathbf{b}_{(01)(3)} = [0\,4\,0\,2\,1\,1\,3\,1\,1\,2]$, $\mathbf{b}_{(01)(4)} = [0\,2\,0\,1\,1\,1\,3\,2\,1\,4]$,
$\mathbf{b}_{(02)(0)} = [0\,2\,0\,2\,1\,3\,3\,0\,1\,3]$, $\mathbf{b}_{(02)(1)} = [0\,2\,0\,3\,1\,0\,3\,3\,1\,2]$, $\mathbf{b}_{(02)(2)} = [0\,3\,0\,0\,1\,3\,3\,2\,1\,2]$,
$\mathbf{b}_{(02)(3)} = [0\,0\,0\,3\,1\,2\,3\,2\,1\,3]$, $\mathbf{b}_{(02)(4)} = [0\,3\,0\,2\,1\,2\,3\,3\,1\,0]$,
$\mathbf{b}_{(03)(0)} = [0\,3\,0\,3\,1\,4\,3\,1\,1\,4]$, $\mathbf{b}_{(03)(1)} = [0\,3\,0\,4\,1\,1\,3\,4\,1\,3]$, $\mathbf{b}_{(03)(2)} = [0\,4\,0\,1\,1\,4\,3\,3\,1\,3]$,
$\mathbf{b}_{(03)(3)} = [0\,1\,0\,4\,1\,3\,3\,3\,1\,4]$, $\mathbf{b}_{(03)(4)} = [0\,4\,0\,3\,1\,3\,3\,4\,1\,1]$,
$\mathbf{b}_{(04)(0)} = [0\,4\,0\,4\,1\,0\,3\,2\,1\,0]$, $\mathbf{b}_{(04)(1)} = [0\,4\,0\,0\,1\,2\,3\,0\,1\,4]$, $\mathbf{b}_{(04)(2)} = [0\,0\,0\,2\,1\,0\,3\,4\,1\,4]$,
$\mathbf{b}_{(04)(3)} = [0\,2\,0\,0\,1\,4\,3\,4\,1\,0]$, $\mathbf{b}_{(04)(4)} = [0\,0\,0\,4\,1\,4\,3\,0\,1\,2]$,
$\mathbf{b}_{(10)(0)} = [1\,0\,1\,0\,2\,1\,4\,3\,2\,1]$, $\mathbf{b}_{(10)(1)} = [1\,0\,1\,1\,2\,3\,4\,1\,2\,0]$, $\mathbf{b}_{(10)(2)} = [1\,1\,1\,3\,2\,1\,4\,0\,2\,0]$,
$\mathbf{b}_{(10)(3)} = [1\,3\,1\,1\,2\,0\,4\,0\,2\,1]$, $\mathbf{b}_{(10)(4)} = [1\,1\,1\,0\,2\,0\,4\,1\,2\,3]$,
$\mathbf{b}_{(11)(0)} = [1\,1\,1\,1\,2\,2\,4\,4\,2\,2]$, $\mathbf{b}_{(11)(1)} = [1\,1\,1\,2\,2\,4\,4\,2\,2\,1]$, $\mathbf{b}_{(11)(2)} = [1\,2\,1\,4\,2\,2\,4\,1\,2\,1]$,
$\mathbf{b}_{(11)(3)} = [1\,4\,1\,2\,2\,1\,4\,1\,2\,2]$, $\mathbf{b}_{(11)(4)} = [1\,2\,1\,1\,2\,1\,4\,2\,2\,4]$,
$\mathbf{b}_{(12)(0)} = [1\,2\,1\,2\,2\,3\,4\,0\,2\,3]$, $\mathbf{b}_{(12)(1)} = [1\,2\,1\,3\,2\,0\,4\,3\,2\,2]$, $\mathbf{b}_{(12)(2)} = [1\,3\,1\,0\,2\,3\,4\,2\,2\,2]$,
$\mathbf{b}_{(12)(3)} = [1\,0\,1\,3\,2\,2\,4\,2\,2\,3]$, $\mathbf{b}_{(12)(4)} = [1\,3\,1\,2\,2\,2\,4\,3\,2\,0]$,
$\mathbf{b}_{(13)(0)} = [1\,3\,1\,3\,2\,4\,4\,1\,2\,4]$, $\mathbf{b}_{(13)(1)} = [1\,3\,1\,4\,2\,1\,4\,4\,2\,3]$, $\mathbf{b}_{(13)(2)} = [1\,4\,1\,1\,2\,4\,4\,3\,2\,3]$,
$\mathbf{b}_{(13)(3)} = [1\,1\,1\,4\,2\,3\,4\,3\,2\,4]$, $\mathbf{b}_{(13)(4)} = [1\,4\,1\,3\,2\,3\,4\,4\,2\,1]$,
$\mathbf{b}_{(14)(0)} = [1\,4\,1\,4\,2\,0\,4\,2\,2\,0]$, $\mathbf{b}_{(14)(1)} = [1\,4\,1\,0\,2\,2\,4\,0\,2\,4]$, $\mathbf{b}_{(14)(2)} = [1\,0\,1\,2\,2\,0\,4\,4\,2\,4]$,
$\mathbf{b}_{(14)(3)} = [1\,2\,1\,0\,2\,4\,4\,4\,2\,0]$, $\mathbf{b}_{(14)(4)} = [1\,0\,1\,4\,2\,4\,4\,0\,2\,2]$,
$\mathbf{b}_{(20)(0)} = [2\,0\,2\,0\,3\,1\,0\,3\,3\,1]$, $\mathbf{b}_{(20)(1)} = [2\,0\,2\,1\,3\,3\,0\,1\,3\,0]$, $\mathbf{b}_{(20)(2)} = [2\,1\,2\,3\,3\,3\,1\,0\,0\,3\,0]$,
$\mathbf{b}_{(20)(3)} = [2\,3\,2\,1\,3\,0\,0\,0\,3\,1]$, $\mathbf{b}_{(20)(4)} = [2\,1\,2\,0\,3\,0\,0\,1\,3\,3]$,
$\mathbf{b}_{(21)(0)} = [2\,1\,2\,1\,3\,2\,0\,4\,3\,2]$, $\mathbf{b}_{(21)(1)} = [2\,1\,2\,2\,3\,4\,0\,2\,3\,1]$, $\mathbf{b}_{(21)(2)} = [2\,2\,2\,4\,3\,2\,0\,1\,3\,1]$,
$\mathbf{b}_{(21)(3)} = [2\,4\,2\,2\,3\,1\,0\,1\,3\,2]$, $\mathbf{b}_{(21)(4)} = [2\,2\,2\,1\,3\,1\,0\,2\,3\,4]$,
$\mathbf{b}_{(22)(0)} = [2\,2\,2\,2\,3\,3\,0\,0\,3\,3]$, $\mathbf{b}_{(22)(1)} = [2\,2\,2\,3\,3\,0\,0\,3\,3\,2]$, $\mathbf{b}_{(22)(2)} = [2\,3\,2\,0\,3\,3\,0\,2\,3\,2]$,
$\mathbf{b}_{(22)(3)} = [2\,0\,2\,3\,3\,2\,0\,2\,3\,3]$, $\mathbf{b}_{(22)(4)} = [2\,3\,2\,2\,3\,2\,0\,3\,3\,0]$,
$\mathbf{b}_{(23)(0)} = [2\,3\,2\,3\,3\,4\,0\,1\,3\,4]$, $\mathbf{b}_{(23)(1)} = [2\,3\,2\,4\,3\,1\,0\,4\,3\,3]$, $\mathbf{b}_{(23)(2)} = [2\,4\,2\,1\,3\,4\,0\,3\,3\,3]$,
$\mathbf{b}_{(23)(3)} = [2\,1\,2\,4\,3\,3\,0\,3\,3\,4]$, $\mathbf{b}_{(23)(4)} = [2\,4\,2\,3\,3\,3\,0\,4\,3\,1]$,
$\mathbf{b}_{(24)(0)} = [2\,4\,2\,4\,3\,0\,0\,2\,3\,0]$, $\mathbf{b}_{(24)(1)} = [2\,4\,2\,0\,3\,2\,0\,0\,3\,4]$, $\mathbf{b}_{(24)(2)} = [2\,0\,2\,2\,3\,0\,0\,4\,3\,4]$,
$\mathbf{b}_{(24)(3)} = [2\,2\,2\,0\,3\,4\,0\,4\,3\,0]$, $\mathbf{b}_{(24)(4)} = [2\,0\,2\,4\,3\,4\,0\,0\,3\,2]$,
$\mathbf{b}_{(30)(0)} = [3\,0\,3\,0\,4\,1\,1\,3\,4\,1]$, $\mathbf{b}_{(30)(1)} = [3\,0\,3\,1\,4\,3\,1\,1\,4\,0]$, $\mathbf{b}_{(30)(2)} = [3\,1\,3\,3\,4\,1\,1\,0\,4\,0]$,
$\mathbf{b}_{(30)(3)} = [3\,3\,3\,1\,4\,0\,1\,0\,4\,1]$, $\mathbf{b}_{(30)(4)} = [3\,1\,3\,0\,4\,0\,1\,1\,4\,3]$,
$\mathbf{b}_{(31)(0)} = [3\,1\,3\,1\,4\,2\,1\,4\,4\,2]$, $\mathbf{b}_{(31)(1)} = [3\,1\,3\,2\,4\,4\,1\,2\,4\,1]$, $\mathbf{b}_{(31)(2)} = [3\,2\,3\,4\,4\,2\,1\,1\,4\,1]$,
$\mathbf{b}_{(31)(3)} = [3\,4\,3\,2\,4\,1\,1\,1\,4\,2]$, $\mathbf{b}_{(31)(4)} = [3\,2\,3\,1\,4\,1\,1\,2\,4\,4]$,
$\mathbf{b}_{(32)(0)} = [3\,2\,3\,2\,4\,3\,1\,0\,4\,3]$, $\mathbf{b}_{(32)(1)} = [3\,2\,3\,3\,4\,0\,1\,3\,4\,2]$, $\mathbf{b}_{(32)(2)} = [3\,3\,3\,0\,4\,3\,1\,2\,4\,2]$,
$\mathbf{b}_{(32)(3)} = [3\,0\,3\,3\,4\,2\,1\,2\,4\,3]$, $\mathbf{b}_{(32)(4)} = [3\,3\,3\,2\,4\,2\,1\,3\,4\,0]$,
$\mathbf{b}_{(33)(0)} = [3\,3\,3\,3\,4\,4\,1\,1\,4\,4]$, $\mathbf{b}_{(33)(1)} = [3\,3\,3\,4\,4\,1\,1\,4\,4\,3]$, $\mathbf{b}_{(33)(2)} = [3\,4\,3\,1\,4\,4\,1\,3\,4\,3]$,
$\mathbf{b}_{(33)(3)} = [3\,1\,3\,4\,4\,3\,1\,3\,4\,4]$, $\mathbf{b}_{(33)(4)} = [3\,4\,3\,3\,4\,3\,1\,4\,4\,1]$,
$\mathbf{b}_{(34)(0)} = [3\,4\,3\,4\,4\,0\,1\,2\,4\,0]$, $\mathbf{b}_{(34)(1)} = [3\,4\,3\,0\,4\,2\,1\,0\,4\,4]$, $\mathbf{b}_{(34)(2)} = [3\,0\,3\,2\,4\,0\,1\,4\,4\,4]$,
$\mathbf{b}_{(34)(3)} = [3\,2\,3\,0\,4\,4\,1\,4\,4\,0]$, $\mathbf{b}_{(34)(4)} = [3\,0\,3\,4\,4\,4\,1\,0\,4\,2]$,
$\mathbf{b}_{(40)(0)} = [4\,0\,4\,0\,0\,1\,2\,3\,0\,1]$, $\mathbf{b}_{(40)(1)} = [4\,0\,4\,1\,0\,3\,2\,1\,0\,0]$, $\mathbf{b}_{(40)(2)} = [4\,1\,4\,3\,0\,1\,2\,0\,0\,0]$,
$\mathbf{b}_{(40)(3)} = [4\,3\,4\,1\,0\,0\,2\,0\,0\,1]$, $\mathbf{b}_{(40)(4)} = [4\,1\,4\,0\,0\,0\,2\,1\,0\,3]$,
$\mathbf{b}_{(41)(0)} = [4\,1\,4\,1\,0\,2\,2\,4\,0\,2]$, $\mathbf{b}_{(41)(1)} = [4\,1\,4\,2\,0\,4\,2\,2\,0\,1]$, $\mathbf{b}_{(41)(2)} = [4\,2\,4\,4\,0\,2\,2\,1\,0\,1]$,
$\mathbf{b}_{(41)(3)} = [4\,4\,4\,2\,0\,1\,2\,1\,0\,2]$, $\mathbf{b}_{(41)(4)} = [4\,2\,4\,1\,0\,1\,2\,2\,0\,4]$,
$\mathbf{b}_{(42)(0)} = [4\,2\,4\,2\,0\,3\,2\,0\,0\,3]$, $\mathbf{b}_{(42)(1)} = [4\,2\,4\,3\,0\,0\,2\,3\,0\,2]$, $\mathbf{b}_{(42)(2)} = [4\,3\,4\,0\,0\,3\,2\,2\,0\,2]$,
$\mathbf{b}_{(42)(3)} = [4\,0\,4\,3\,0\,2\,2\,2\,0\,3]$, $\mathbf{b}_{(42)(4)} = [4\,3\,4\,2\,0\,2\,2\,3\,0\,0]$,
$\mathbf{b}_{(43)(0)} = [4\,3\,4\,3\,0\,4\,2\,1\,0\,4]$, $\mathbf{b}_{(43)(1)} = [4\,3\,4\,4\,0\,1\,2\,4\,0\,3]$, $\mathbf{b}_{(43)(2)} = [4\,4\,4\,1\,0\,4\,2\,3\,0\,3]$,
$\mathbf{b}_{(43)(3)} = [4\,1\,4\,4\,0\,3\,2\,3\,0\,4]$, $\mathbf{b}_{(43)(4)} = [4\,4\,4\,3\,0\,3\,2\,4\,0\,1]$,
$\mathbf{b}_{(44)(0)} = [4\,4\,4\,4\,0\,0\,2\,2\,0\,0]$, $\mathbf{b}_{(44)(1)} = [4\,4\,4\,0\,0\,2\,2\,0\,0\,4]$, $\mathbf{b}_{(44)(2)} = [4\,0\,4\,2\,0\,0\,2\,4\,0\,4]$,
$\mathbf{b}_{(44)(3)} = [4\,2\,4\,0\,0\,4\,2\,4\,0\,0]$, $\mathbf{b}_{(44)(4)} = [4\,0\,4\,4\,0\,4\,2\,0\,0\,2]$,

4. An interleaving construction for GPFs. We now describe a method which enables us to construct many new GPFs; it is similar to Construction 3.1, and is actually a generalization of Construction 3.1 of [9].

4.1. The construction method.

CONSTRUCTION 4.1. *Suppose c, n, t, and v are positive integers where $c \geq 2$. Suppose also that*

$$A = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{c^v/n-1}\}$$

is an (n, c, v)–PF. Consider the set S of all n-ary cycles $\mathbf{x} = [x_0, x_1, \ldots, x_{t-1}]$ with the property that

$$\sum_{i=0}^{t-1} x_i \equiv 1 \pmod{n}.$$

If $\mathbf{x}, \mathbf{y} \in S$, then write $\mathbf{x} \sim \mathbf{y}$ if and only if $\mathbf{x} = E^i(\mathbf{y})$ for some i. It is simple to verify that $\sim$ is an equivalence relation on S which partitions S into q classes, say. Now, let

$$X = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{q-1}\}$$

be a set of $\sim$-class representatives. Next, let

$$A^t = \{(\mathbf{a}_{i_0}, \mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_{t-1}}) \,:\, \mathbf{a}_{i_0}, \mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_{t-1}} \in A\}$$

be the set of all t-tuples of elements of A. Now, define B' to be the collection of all cycles of the form

$$\mathcal{I}(E^0\mathbf{a}_{i_0}, E^{x_0}\mathbf{a}_{i_1}, E^{x_0+x_1}\mathbf{a}_{i_2}, \ldots, E^{x_0+x_1+\cdots+x_{t-2}}\mathbf{a}_{i_{t-1}}),$$

where $(x_0, x_1, \ldots, x_{t-1}) \in X$ and $(\mathbf{a}_{i_0}, \mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_{t-1}}) \in A^t$. Hence, $|B'| = qc^{tv}/n^t$.

Finally, put $B = \{\mathcal{T}(\mathbf{z}) \,:\, \mathbf{z} \in B'\}$. Note that while B' may contain duplicate cycles, B (defined as a set) will not, i.e., duplicates are discarded.

We can now state and prove the following result.

THEOREM 4.2. *Suppose c, n, t, v, and A satisfy the conditions of Construction* 4.1*. If B is constructed from A using Construction* 4.1*, then B is a collection of cycles with the property that every c-ary (tv)-tuple occurs exactly once in a cycle of B. Every cycle $\mathbf{b} \in B$ has least period $\ell_{\mathbf{b}}n$, for some positive integer $\ell_{\mathbf{b}}$ satisfying $\ell_{\mathbf{b}} | t$ and $(\frac{t}{\ell_{\mathbf{b}}}, n) = 1$.*

Proof. Suppose $\mathbf{y}$ is any c-ary (tv)-tuple. We first show that $\mathbf{y}$ occurs in one of the cycles of B'. Suppose

$$\mathbf{y} = \mathcal{I}(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}),$$

where $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$ are c-ary t-tuples. Now suppose that $\mathbf{z}_i$ occurs in cycle $\mathbf{a}_{\ell_i}$ at position k_i, for every i satisfying $0 \le i < t$. In addition we define a further n-ary t-tuple $\mathbf{x} = (x_0, x_1, \ldots, x_{t-1})$ where $x_i \equiv k_{i+1} - k_i \pmod n$, for every i satisfying $0 \le i < t-1$, and $x_{t-1} \equiv k_0 - k_{t-1} + 1 \pmod n$. First observe that $\mathbf{x} \in S$, since

$$\sum_{i=0}^{t-1} x_i \equiv \sum_{i=0}^{t-2}(k_{i+1} - k_i) + (k_0 - k_{t-1} + 1) \equiv 1 \pmod n.$$

Hence, there exists some cyclic shift of $\mathbf{x}$, say

$$E^u(\mathbf{x}) = (x_u, x_{u+1}, \ldots, x_{t-1}, x_0, \ldots, x_{u-1}),$$

which is a member of X. Hence, if we define the n-ary t-tuple $(v_0, v_1, \ldots, v_{t-1})$ by

$$v_i = \begin{cases} 0 & \text{if } i = 0, \\ \sum_{j=u}^{i+u-1} x_j \bmod n & \text{(subscripts modulo } t\text{) if } 0 < i \le t-1, \end{cases}$$

then the following cycle is a member of B':

$$\mathbf{w} = \mathcal{I}(E^{v_0}\mathbf{a}_{\ell_u}, E^{v_1}\mathbf{a}_{\ell_{u+1}}, \ldots, E^{v_{t-u-1}}\mathbf{a}_{\ell_{t-1}}, E^{v_{t-u}}\mathbf{a}_{\ell_0}, \ldots, E^{v_{t-1}}\mathbf{a}_{\ell_{u-1}}).$$

Now, $\mathbf{z}_{u+i}$ occurs in $E^{v_i}(\mathbf{a}_{\ell_{u+i}})$ at position $k_{u+i} - v_i$, $(0 \le i < t-u)$, and z_i occurs in $E^{v_{i+t-u}}(\mathbf{a}_{\ell_i})$ at position $k_i - v_{t-u+i}$, $(0 \le i \le u-1)$, where positions are calculated modulo n. By definition of $\mathbf{x}$ we also have

$$v_i = \begin{cases} 0 & \text{if } i = 0, \\ k_{u+i} - k_u \bmod n & \text{if } 0 < i < t-u, \\ k_{u-t+i} - k_u + 1 \bmod n & \text{if } t-u \le i \le t-1. \end{cases}$$

Thus, $\mathbf{z}_{u+i}$ occurs in $E^{v_i}(\mathbf{a}_{\ell_{u+i}})$ at position k_u, $(0 \le i < t-u)$, and $\mathbf{z}_i$ occurs in $E^{v_{i+t-u}}(\mathbf{a}_{\ell_i})$ at position $k_u + 1$, $(0 \le i \le u-1)$. Hence, $\mathbf{y}$ occurs in $\mathbf{w}$ at position $k_u t - u$.

Now, since a (tv)-tuple $\mathbf{y}$ occurs in a cycle of B', it follows (from the way in which B was derived from B') that $\mathbf{y}$ must occur in a cycle of B. Next, suppose that $\mathbf{y}$ occurs at two different points in the cycles of B'. Now, because A is a PF, $\mathbf{y}$ can only arise from one $(t-1)$-tuple of "relative shifts" and one t-tuple from A^t. Hence $\mathbf{y}$ can only arise twice if the same $(t-1)$-tuple of relative shifts occurs twice in the same element of X (the same $(t-1)$-tuple of relative shifts cannot arise in different elements of X since X contains a unique element from each equivalence class under $\sim$ and this class is uniquely determined by a $(t-1)$-tuple of relative shifts). That is, the same (tv)-tuple can only occur multiple times in two ways:

- within the same cycle of B', or
- in two distinct cycles of B' generated by the same set of relative shifts $\mathbf{x}$ and by two different cyclic shifts of the same t-tuple of elements of A.

In both cases this can only happen when the t-tuple of relative shifts used to derive the cycle(s) ($\mathbf{x}$ say) satisfies $\mathbf{x} = E^i\mathbf{x}$ for some i $(0 < i < t)$. The second case is rather easier to deal with, since in this case the resulting cycles of B' will be identical to one another (except for a cyclic shift). Hence, the duplication will be removed when B is derived from B'. We therefore need only consider the first case. If the same (tv)-tuple occurs twice within the same cycle $\mathbf{b}$ of B', say at positions i and j, then we must have $E^i\mathbf{b} = E^j\mathbf{b}$, and hence the (tv)-tuple will *not* be repeated within $\mathcal{T}(\mathbf{b})$. Hence, all the (tv)-tuples in the cycles of B are distinct.

We next consider the possible periods of the cycles in B. Suppose $\mathbf{b} = E^i\mathbf{b}$ for some i $(0 < i \le nt)$. Note that we must have $i | nt$. Suppose also that $i' = i \bmod t$, and hence if $\mathbf{x} \in X$ is used to produce $\mathbf{b}$, then $\mathbf{x} = E^{i'}\mathbf{x}$ and so $i' | t$. Now, by definition of S, if $\mathbf{x} = [x_0, x_1, \ldots, x_{t-1}]$, then $\sum_{j=0}^{t-1} x_j \equiv 1 \bmod n$, and hence, since $\mathbf{x} = E^{i'}(\mathbf{x})$, we have

$$\left(\frac{t}{i'}\right) \sum_{j=0}^{i'-1} x_j \equiv 1 \pmod{n}.$$

Note that this implies that $(t/i', n) = 1$ and also that $(\sum_{j=0}^{i'-1} x_j, n) = 1$.

Now, since $i' | t$ and $i \equiv i' \pmod{t}$, it follows that $i' | i$, say $i = \nu i'$. Hence, since $\mathbf{b} = E^i\mathbf{b}$, we have

$$\nu \sum_{j=0}^{i'-1} x_j \equiv 0 \pmod{n}$$

(this follows since the total relative shift at a displacement of i in $\mathbf{b}$ must be zero). But we have already observed that $(\sum_{j=0}^{i'-1} x_j, n) = 1$, and hence we must have $n | \nu$.

Hence, $ni'|i$ and $(t/i', n) = 1$. Since we have already observed that $i|nt$, the desired result on the periods of cycles in B follows. □

When we combine the above result with Theorem 1.21, we immediately have the following.

COROLLARY 4.3. *If an (n, c, v)–PF exists, then there exists a (n, t, c, tv)–GPF for every positive integer t.*

REMARK 4.4. *In fact the cycles of the GPF in this corollary can be derived directly from the cycles in the set B' of Construction* 4.1 *merely by discarding duplicate cycles from the set (that is, without truncating cycles as in the derivation of B from B'). This means that each cycle in the GPF is obtained by t-fold interleaving of the cycles of the (n, c, v)–PF.*

REMARK 4.5. *It is straightforward to see that $n|t^{n-1}$ if and only if $(t/\ell, n) \neq 1$ for every factor ℓ of t (except for $\ell = t$). Hence, if $n|t^{n-1}$, then Construction* 4.1 *yields a set B of cycles of period exactly nt (in fact $B = B'$), and hence B is an (nt, c, tv)–PF. This corresponds to Construction* 3.1 *of* [9].

4.2. Examples.

EXAMPLE 4.6. *Let A be the $(5, 5, 1)$–PF consisting of the single cycle* [01234]. *Then, to apply Construction* 4.1 *to this cycle with $t = 3$, we first need to define*

$$X = \{[001], [024], [033], [042], [114], [123], [132], [222], [344]\}.$$

Using this choice for X we then obtain the following set B of nine cycles (of periods 15 *and* 5*) in which every* 5*-ary* 3*-tuple occurs exactly once.*

$$\begin{array}{lll} [0\,0\,0\,1\,1\,1\,2\,2\,2\,3\,3\,3\,4\,4\,4], & [0\,0\,2\,1\,1\,3\,2\,2\,4\,3\,3\,0\,4\,4\,1], & [0\,0\,3\,1\,1\,4\,2\,2\,0\,3\,3\,1\,4\,4\,2], \\ [0\,0\,4\,1\,1\,0\,2\,2\,1\,3\,3\,2\,4\,4\,3], & [0\,1\,2\,1\,2\,3\,2\,3\,4\,3\,4\,0\,4\,0\,1], & [0\,1\,3\,1\,2\,4\,2\,3\,0\,3\,4\,1\,4\,0\,2], \\ [0\,1\,4\,1\,2\,0\,2\,3\,1\,3\,4\,2\,4\,0\,3], & [0\,2\,4\,1\,3], & [0\,3\,2\,1\,4\,3\,2\,0\,4\,3\,1\,0\,4\,2\,1]. \end{array}$$

Using Theorem 1.21*, the set B can be used to produce a $(5, 3, 5, 3)$–GPF.*

EXAMPLE 4.7. *Let A be the following set of five* 5*-ary cycles of period* 5*, which constitute a $(5, 5, 2)$–PF.*

$$\mathbf{a}_0 = [0\,0\,1\,3\,1], \mathbf{a}_1 = [1\,1\,2\,4\,2], \mathbf{a}_2 = [2\,2\,3\,0\,3], \mathbf{a}_3 = [3\,3\,4\,1\,4], \mathbf{a}_4 = [4\,4\,0\,2\,0].$$

Put $t = 2$ and

$$X = \{[33], [01], [42]\}.$$

In the table below, we give the set of 65 *cycles resulting from applying Construction* 4.1 *to A with $t = 2$. In each row we give the three cycles obtained by applying the three "shift tuples" of X to a pair of interleaved cycles from A, with indices as marked at the start of the row. Note that the* 10 *duplicate cycles (which do not count as part of the* 65 *cycles) are preceded with an asterisk, and arise when the representative from X has cyclic symmetry. Five of the cycles have period* 5 *and sixty have period* 10*, and hence we can use these cycles to produce a $(5, 2, 5, 4)$–GPF.*

	[33]	[01]	[42]
00	[03011]	[0000113311]	[0100103113]
01	[0402113112],	[0101123412],	[0201113214],
02	[0003123213],	[0202133013],	[0302123310],
03	[0104133314],	[0303143114],	[0403133411],
04	[0200143410],	[0404103210],	[0004143012],
10	∗[1311204021],	[1010214321],	[1110204123],
11	[14122],	[1111224022],	[1211214224],
12	[1013224223],	[1212234023],	[1312224320],
13	[1114234324],	[1313244124],	[1413234421],
14	[1210244420],	[1414204220],	[1014244022],
20	∗[2321300031],	[2020310331],	[2120300133],
21	∗[2422310132],	[2121320032],	[2221310230],
22	[20233],	[2222330033],	[2322320330],
23	[2124330334],	[2323340134],	[2423330431],
24	[2220340430],	[2424300230],	[2024340032],
30	∗[3331401041],	[3030411341],	[3130401143],
31	∗[3432411142],	[3131421442],	[3231411244],
32	∗[3033421243],	[3232431043],	[3332421340],
33	[31344],	[3333441144],	[3433431441],
34	[3230441440],	[3434401240],	[3034441042],
40	∗[4341002001],	[4040012301],	[4140002103],
41	∗[4442012102],	[4141022402],	[4241012204],
42	∗[4043022203],	[4242032003],	[4342022300],
43	∗[4144032304],	[4343042104],	[4443032401],
44	[42400],	[4444002200],	[4044042002].

5. The Lempel homomorphism and the construction of PMFs and GPFs. The Lempel homomorphism [4] (and its generalization to arbitrary finite fields), has been very widely applied in the construction of de Bruijn sequences [4], Perfect Factors [1, 12] and Perfect Maps [13]. We now briefly show how it can be applied to the construction of PMFs and GPFs over alphabets Z_c.

5.1. The Lempel homomorphism. We first define a version of the Lempel homomorphism on c-ary cycles, where the elements of the c-ary alphabet are taken as the integers modulo c.

DEFINITION 5.1. *We define the Lempel homomorphism D acting on c-ary cycles to be the operator $E-1$ (we will usually write $E-1$ for D). Thus if c, n are positive integers ($c > 1$), and $\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$ is a c-ary cycle of period n, then $D\mathbf{a}$ is the following c-ary cycle of period n*

$$[a_1 - a_0, a_2 - a_1, \ldots, a_{n-1} - a_{n-2}, a_0 - a_{n-1}],$$

where the arithmetic is computed modulo c.

DEFINITION 5.2. *Suppose c, n are positive integers ($c > 1$), and let*

$$\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$$

be a c-ary cycle of period n and weight w. Then we define the pre-image *of $\mathbf{a}$ under D, denoted $D^{-1}\mathbf{a}$ or $(E-1)^{-1}\mathbf{a}$, to be the following set of (w, c) c-ary cycles of period*

$nc/(w,c)$:

$$\left\{ [s, s+a_0, s+a_0+a_1, \ldots, s+\sum_{i=0}^{n-2} a_i, s+w, s+w+a_0, s+w+a_0+a_1, \ldots, s+w+\sum_{i=0}^{n-2} a_i, s+2w, \ldots, s+(c/(w,c)-1)w+\sum_{i=0}^{n-2} a_i] \ : \ 0 \le s < (w,c) \right\}.$$

Clearly, $\mathbf{a} \in D^{-1}D\mathbf{a}$ *for any cycle* $\mathbf{a}$. *We call the operator* $(E-1)^{-1}$ *the* Lempel inverse homomorphism (LIH).

Of course, given a cycle $\mathbf{a}$ as in the above definition, we can apply $(E-1)^{-1}$ to the set $(E-1)^{-1}\mathbf{a}$ to obtain a second set of cycles, which we denote by $(E-1)^{-2}\mathbf{a}$. Notice that the cycles of this set need not all have the same period (because the cycles in $(E-1)^{-1}\mathbf{a}$ need not all have the same weight). We can continue in this way and write $(E-1)^{-k}\mathbf{a}$ for the set of cycles obtained by making k applications of $(E-1)^{-1}$ to $\mathbf{a}$.

We also need to define the action of the Lempel homomorphism and its inverse on c-ary tuples. For convenience we also denote these mappings by D and D^{-1} (the domain of the mapping should always be clear from the context).

DEFINITION 5.3. *Suppose* c *and* v *are positive integers* $(c > 1)$, *and let*

$$\mathbf{s} = (s_0, s_1, \ldots, s_{v-1})$$

be a c*-ary* v*-tuple. Then define* $D\mathbf{s}$ *to be the following* c*-ary* $(v-1)$*-tuple*:

$$(s_1 - s_0, s_2 - s_1, \ldots, s_{t-1} - s_{t-2}).$$

On the other hand if $\mathbf{w} = (w_0, w_1, \ldots, w_{v-2})$ *is a* c*-ary* $(v-1)$*-tuple, then we define* $D^{-1}\mathbf{w}$ *to be the following* c*-set of* c*-ary* v*-tuples*:

$$D^{-1}\mathbf{w} = \left\{ (s, s+w_0, s+w_0+w_1, \ldots, s+\sum_{i=0}^{v-2} w_i) : s \in Z_c \right\}.$$

We will also use $E-1$ *and* $(E-1)^{-1}$ *to denote* D *and* D^{-1} *acting on* c*-ary tuples.*

We can now state the following result which follows immediately from the definitions.

LEMMA 5.4. *Let* $\mathbf{a}$ *be a* c*-ary cycle of period* n, $\mathbf{s}$ *a* c*-ary* v*-tuple and* $\mathbf{w}$ *a* c*-ary* $(v-1)$*-tuple. Then*

- $D\mathbf{s} = \mathbf{w}$ *if and only if* $\mathbf{s} \in D^{-1}\mathbf{w}$,
- *if* $\mathbf{s}$ *appears in* $\mathbf{a}$ *at position* p, *then* $D\mathbf{s}$ *appears in* $D\mathbf{a}$ *at position* p, *and*
- *if* $\mathbf{s}$ *appears in* $\mathbf{a}$ *at position* p, *then any* $(v+1)$*-tuple of* $D^{-1}\mathbf{s}$ *appears in some cycle of* $(E-1)^{-1}\mathbf{a}$ *at a position* p' *with* $p' \equiv p \bmod n$.

We use the following construction method, which is based on the Lempel inverse homomorphism, to construct Perfect Factors, PMFs, and GPFs.

CONSTRUCTION 5.5. *Suppose* c *and* r *are positive integers, where* $c > 1$, *and let* A *be a set of* c*-ary cycles*

$$\{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{t-1}\}$$

of periods $\ell_0, \ell_1, \ldots, \ell_{t-1}$ *and weights* $w_0, w_1, \ldots, w_{t-1}$, *respectively. Then let* B *be the following set of* $\sum_{i=0}^{t-1}(w_i, c)$ *cycles*:

$$B = \bigcup_{i=0}^{t-1} (E-1)^{-1}\mathbf{a}_i.$$

We now have the following.

THEOREM 5.6. *Suppose c, m, n, and v are positive integers ($c > 1$), and let A be a set of c-ary cycles of constant weight w. Suppose also that B is derived from A using Construction* 5.5. *Then*

- *if A is an (n, c, v)–PF, then B is an $(nc/(w, c), c, v + 1)$–PF,*
- *if A is an (m, n, c, v)–PMF, then B is an $(mc/(w, c), n, c, v + 1)$–PMF, and*
- *if A is an (m, n, c, v)–GPF, then B is an $(mc/(w, c), n, c, v + 1)$–GPF.*

Proof. This follows immediately from the definition of the Lempel inverse homomorphism and Lemma 5.4. □

By considering the special case where $w = 0$, we immediately have the following.

COROLLARY 5.7. *Suppose c, m, n, and v are positive integers ($c > 1$), and let A be a set of c-ary cycles of constant weight zero. Suppose also that B is derived from A using Construction* 5.5. *Then*

- *if A is an (n, c, v)–PF, then B is an $(n, c, v + 1)$–PF,*
- *if A is an (m, n, c, v)–PMF, then B is an $(m, n, c, v + 1)$–PMF, and*
- *if A is an (m, n, c, v)–GPF, then B is an $(m, n, c, v + 1)$–GPF.*

Of course, if the set of cycles B in Theorem 5.6 or Corollary 5.7 has constant weight, then Construction 5.5 can be applied again to B to produce a new set of cycles which will again form a Perfect Factor/PMF/GPF. This process can be repeated to produce a series of Perfect Factors/PMFs/GPFs with increasing window size, so long as the cycles in each set all have the same weight. In the next section we will see how the interleaving constructions of sections 3 and 4 can be combined with repeated use of Construction 5.5 to produce a powerful set of construction methods.

5.2. Examples.

EXAMPLE 5.8. *The $(5, 2, 5, 4)$–PMF constructed in Example* 3.3 *has constant weight zero. Hence, if we apply Construction* 5.5 *then, by Corollary* 5.7*, we obtain a $(5, 2, 5, 5)$–PMF.*

EXAMPLE 5.9. *The $(5, 2, 5, 4)$–GPF constructed in Example* 4.7 *has constant weight zero. Hence, if we apply Construction* 5.5 *then, by Corollary* 5.7*, we obtain a $(5, 2, 5, 5)$–GPF.*

6. Combining interleaving and the Lempel homomorphism. Consider applying one of Constructions 3.1 or 4.1 to an (n, c, v)–PF A. The resulting set of cycles B will be either a (n, t, c, tv)–PMF, a (n, t, c, tv)–GPF, or a (tn, c, tv)–PF. We ask: what is the maximum number of times that Construction 5.5 can be applied to the cycles of B while yielding a set of cycles of period tn? In order for the construction to be applicable δ times, we require that the set

$$\{(E - 1)^{-(\delta-1)}\mathbf{b}, \mathbf{b} \in B\}$$

be constant weight zero. By repeated use of Corollary 5.7, it follows that if δ applications are possible while maintaining zero weight, then we can obtain either an $(n, t, c, tv + \delta)$–PMF, an $(n, t, c, tv + \delta)$–GPF, or a $(tn, c, tv + \delta)$–PF.

The answer to our question depends on the maximum value of k such that the set

$$\{(E - 1)^{-k}\mathbf{a}, \mathbf{a} \in A\}$$

is constant weight, as well as on the prime factorizations of t and c. Before giving the answer, we need some preliminary results.

LEMMA 6.1. *Suppose that c does not divide t. Then in $Z_c[E]$, $E - 1$ divides $E^t - 1$ exactly once.*

Proof. In $Z_c[E]$, we have $E^t - 1 = (E-1)g_t(E)$, where

$$g_t(E) := E^{t-1} + E^{t-2} + \cdots + E + 1$$

satisfies $g_t(1) = t$. When c does not divide t, we have $g_t(1) \neq 0 \bmod c$ and so $E-1$ does not divide $g_t(E)$. The lemma follows. □

LEMMA 6.2. *Suppose that c is square-free (i.e., c is a product of distinct primes). Let $t = \prod_i {p_i}^{\beta_i}$ and $c = \prod_i {p_i}^{\alpha_i}$, where $\beta_i \geq 0$ and $\alpha_i = 0$ or 1, be the prime factorizations of t and c. Then in $Z_c[E]$, $E-1$ divides $E^t - 1$ exactly $\delta_{t,c}$ times, where*

$$\delta_{t,c} = \min_{\alpha_i = 1} \left\{ {p_i}^{\beta_i} \right\}.$$

Proof. Consider first the case where c is a prime p and $t = p^\beta$. Then in $Z_p[E]$,

$$E^t - 1 = E^{p^\beta} - 1 = (E-1)^{p^\beta},$$

since $\binom{p^\beta}{i} = 0 \bmod p$ for $1 \leq i \leq p^\beta - 1$. So in this case, $E-1$ divides $E^t - 1$ exactly $t = p^\beta$ times.

Now let t and c have prime factorizations as in the statement of the lemma. Suppose $\alpha_i = 1$. Then in $Z_{p_i}[E]$,

$$E^t - 1 = (E^{{p_i}^{\beta_i}} - 1)(E^{(\ell-1){p_i}^{\beta_i}} + \cdots + E^{{p_i}^{\beta_i}} + 1),$$

where $\ell = t/{p_i}^{\beta_i}$ is coprime to p_i. So in $Z_{p_i}[E]$,

$$E^t - 1 = (E-1)^{{p_i}^{\beta_i}} g_\ell(E^{{p_i}^{\beta_i}}).$$

But $g_\ell(1) = \ell \neq 0 \bmod p_i$, so we deduce that in $Z_{p_i}[E]$, $E-1$ divides $E^t - 1$ exactly ${p_i}^{\beta_i}$ times. But, by a Chinese Remainder Theorem argument, $E-1$ divides $E^t - 1$ at least δ times in $Z_c[E]$ if and only if it does so at least δ times over each polynomial ring $Z_{p_i}[E]$ for which p_i divides c. The result follows. □

Now suppose A is an (n, c, v)–PF and that for some $w \in \mathbf{Z}_c$, some $k \geq 0$ and for each $\mathbf{a} \in A$, any cycle in $(E-1)^{-k}\mathbf{a}$ has period n and weight w. Then each $\mathbf{a} \in A$ satisfies

$$\frac{E^n - 1}{E - 1}(E-1)^{-k}\ \mathbf{a} = [w, w, \ldots, w]. \tag{6.1}$$

If $w = 0$, then this means that Corollary 5.7 can be applied up to $k+1$ times to the cycles of A to produce $(n, c, v+\delta)$–PFs for each $1 \leq \delta \leq k+1$. If $w \neq 0$, then we have

$$\frac{E^n - 1}{E - 1}(E-1)^{-(k-1)}\ \mathbf{a} = (E-1)[w, w, \ldots, w] = [0, 0, \ldots, 0],$$

and we see that up to k applications of Construction 5.5 to A are possible to produce $(n, c, v+\delta)$–PFs for each $1 \leq \delta \leq k$. Theorem 5.6 guarantees that a final application of Construction 5.5 can be used to yield a $(nc/(w,c), c, v+k+1)$–PF.

Now let B be obtained from A by t-fold interleaving, either as in Construction 3.1 (to obtain a PMF) or as in Construction 4.1 combined with Theorem 1.21 (to obtain a GPF). Then in either case (and by Remark 4.4 in the second case), each cycle of B satisfies relation (6.1) but with E replaced by E^t, i.e., if $\mathbf{b} \in B$, then

$$\frac{E^{tn} - 1}{(E^t - 1)^{(k+1)}}\ \mathbf{b} = [w, w, \ldots, w].$$

Writing $E^t - 1 = (E-1)^{\delta_{t,c}} h_{t,c}(E)$ where $h_{t,c}(E)$ is not divisible by $E-1$, we have, for each $\mathbf{b} \in B$,

$$\frac{E^{tn}-1}{(E-1)^{(k+1)\delta_{t,c}}} \cdot \frac{1}{h_{t,c}(E)^{k+1}}\ \mathbf{b} = [w, w, \ldots, w].$$

Multiplying by $h_{t,c}(E)^{k+1}$, and noting that $h_{t,c}(E)^{k+1}[w]$ is also a constant cycle, we see that for some w' (where $w' = 0$ if $w = 0$),

$$\frac{E^{tn}-1}{E-1} \cdot (E-1)^{-((k+1)\delta_{t,c}-1)}\ \mathbf{b} = [w', w', \ldots, w'], \qquad \mathbf{b} \in \mathbf{B}.$$

We can interpret this equation as follows. If $w' = 0$ (in particular, if $w = 0$), then the sequences of the set

$$\{(E-1)^{-((k+1)\delta_{t,c}-1)}\mathbf{b}, \mathbf{b} \in B\}$$

have zero weight and period tn, so that Construction 5.5 can be applied up to $(k+1)\delta_{t,c}$ times to the cycles of B. Similarly, if $w' \neq 0$, then Construction 5.5 can be applied up to $(k+1)\delta_{t,c} - 1$ times to the cycles of B.

We summarize with the following theorem.

THEOREM 6.3. *Suppose c is square-free. Let B be a (tn, c, tv)–PF/(n, t, c, tv)–PMF/ (n, t, c, tv)–GPF obtained from (n, c, v)–PF A by t-fold interleaving. Suppose that Construction* 5.5 *applied $k \geq 0$ times to the cycles of A results in cycles of period n all having weight w. We write $\ell = (k+1)\delta_{t,c}$. If $w = 0$, then Construction* 5.5 *can be applied up to ℓ times to the cycles of B, resulting in constant weight $(tn, c, tv+\delta)$–PFs/$(n, t, c, tv+\delta)$–PMFs/ $(n, t, c, tv+\delta)$–GPFs for each $1 \leq \delta \leq \ell$. If $w \neq 0$, then Construction* 5.5 *can be applied up to $\ell - 1$ times to the cycles of B, resulting in constant weight $(tn, c, tv+\delta)$–PFs/$(n, t, c, tv+\delta)$–PMFs/ $(n, t, c, tv+\delta)$–GPFs for each $1 \leq \delta \leq \ell - 1$.*

EXAMPLE 6.4. *Let A be the $(5, 5, 2)$–PF of Example* 3.3. *It is easy to verify that the sequences of A satisfy*

$$(E-1)^2\mathbf{a} = [1], \qquad \mathbf{a} \in A.$$

Over Z_5, we have $E^5 - 1 = (E-1)^5$, and so we can write

$$\frac{E^5-1}{E-1}(E-1)^{-2}\mathbf{a} = (E-1)^2\mathbf{a} = [1], \qquad \mathbf{a} \in A,$$

and we can take $k = 2$ and $w = 1$ in Theorem 6.3. *Applying Theorem* 3.2 *with $t = 2$, we can construct a $(5, 2, 5, 4)$–PMF B. Now $\delta_{2,5} = 1$, so according to Theorem* 6.3, *Corollary* 5.7 *can be applied up to $l - 1 = 2$ times to the cycles of B, resulting in a constant weight $(5, 2, 5, 5)$–PMF and a constant weight $(5, 2, 5, 6)$–PMF.*

EXAMPLE 6.5. *The $(5, 3, 5, 3)$–GPF constructed in Example* 4.6 *was obtained from the $(5, 5, 1)$–PF consisting of the single cycle* [01234]. *Arguing as in the above example, we can take $k = 3$ and $w = 1$ in Theorem* 6.3 *to see that Construction* 5.5 *can be applied up to $l-1 = 3$ times to the cycles of the GPF, resulting in a constant weight $(5, 3, 5, 4)$–GPF, a constant weight $(5, 3, 5, 5)$–GPF, and a constant weight $(5, 3, 5, 6)$–GPF.*

7. Perfect Factors for small windows.

7.1. A reduction for the existence problem. Corollary 1.26 allows us to make an important reduction in the sets of parameters for which we need to consider the existence question for Perfect Factors.

Recall from the discussion in section 1.2.2 that to prove Conjecture 1.4 for any fixed v, we need only construct Perfect Factors with parameters (n, c, v) $(n > v + 1)$, where

$$c = \prod_{i=1}^{t} {p_i}^{r_i} \quad \text{and} \quad n = \prod_{i=1}^{t} {p_i}^{s_i},$$

and both $0 \leq s_i \leq r_i v$ and ${p_i}^{s_i} \leq v$ for each i.

For a particular choice of c and v as above, we write

$$c' = \prod_{s_i \neq 0} p_i.$$

Now for each i, ${p_i}^{s_i} \leq v \leq {p_i}^{v}$. Hence $s_i \leq v$ and so $n|(c')^v$. Thus the parameters (n, c', v) satisfy the necessary conditions of Lemma 1.3. Moreover, by Corollary 1.26, the existence of such an (n, c', v)–PF implies the existence of a (n, c, v)–PF. So to settle Conjecture 1.4 for v, it is sufficient to construct Perfect Factors for all parameters (n, c, v) where $n > v + 1$, $c = p_1 \dots p_t$ is square-free, and where $n = \prod_{i=1}^{t} {p_i}^{s_i}$ with $1 \leq s_i$ and ${p_i}^{s_i} \leq v$ for each i.

Notice this means that every prime p_i that divides c must in turn divide n. Moreover, each p_i satisfies $p_i \leq v$. So to settle the existence question for any particular v, it is sufficient to consider Perfect Factors for a finite set of alphabets (whose sizes are products of distinct primes) and for a small set of parameters for each of these alphabets.

We summarize the above reduction formally as the following.

LEMMA 7.1. *Suppose $v \geq 1$ is fixed, and that there exist (n, c, v)–PFs for every square-free $c = p_1 \dots p_t$ and every $n > v + 1$ with $n = \prod_{i=1}^{t} {p_i}^{s_i}$ where $s_i \geq 1$ and ${p_i}^{s_i} \leq v$ for each i. Then Conjecture* 1.4 *is true for v.*

REMARK 7.2. *Note that, because $v < n$, t is always at least* 2 *in the above lemma.*

7.2. Perfect Factors for $v \leq 6$. We now show that Conjecture 1.4 is true for $v \leq 6$. This has already been shown for $v \leq 4$. However, in order to demonstrate the power of our new construction methods, we consider anew all v up to $v = 6$.

7.2.1. Perfect Factors for $v = 2$. For $v = 2$, there is no parameter set satisfying the conditions of Lemma 7.1. We conclude that Conjecture 1.4 is true for $v = 2$. In fact, this means that the methods of [7] are strong enough to settle the existence problem in this case, as already noted in the introductory section.

7.2.2. Perfect Factors for $v = 3$. By Lemma 7.1, we need only consider the existence of a $(6, 6, 3)$–PF. A Perfect Factor with these parameters was obtained in Example 2.6.

7.2.3. Perfect Factors for $v = 4$. Again by Lemma 7.1, only the following two parameter sets need to be considered: $(6, 6, 4)$ and $(12, 6, 4)$.

A PF for the first parameter set was obtained in Example 2.10. A $(12, 6, 4)$–PF can be obtained by applying Construction 4.1 to a $(6, 6, 2)$–PF with $t = 2$ (see Remark 4.5).

7.2.4. Perfect Factors for $v = 5$. By Lemma 7.1, only the following six parameter sets need to be considered:

$$(10, 10, 5), (12, 6, 5), (15, 15, 5), (20, 10, 5), (30, 30, 5), \text{ and } (60, 30, 5).$$

The parameter sets $(10, 10, 5)$, $(20, 10, 5)$, $(30, 30, 5)$, and $(60, 30, 5)$ fall to Theorem 2.7.

Consider the parameters $(12, 6, 5)$. The polynomial $X^{12} - 1$ factorizes as $(X + 1)^4(X^2 + X + 1)^4$ in $Z_2[X]$ and as $(X - 1)^3(X^3 + X^2 + X + 1)^3$ in $Z_3[X]$. We take $g(X) = (X + 1)(X^2 + X + 1)^3$, $p = 2$, and $r = l = 2$ in Construction 2.4 to obtain a $(4, 3, 2, 5)$–GPF. Similarly, we take $g(X) = (X-1)(X^3+X^2+X+1)^2$, $p = 3$, and $r = l = 1$ in Construction 2.4 to obtain a $(3, 4, 3, 5)$–GPF. Combining these GPFs using Construction 1.13, we obtain (according to Theorem 1.22) a $(12, 6, 5)$–PF.

Finally, consider the parameters $(15, 15, 5)$. By considering the factorization of $X^{15} - 1$ in $Z_3[X]$ and $\mathrm{Z}_5[X]$ and following a similar procedure to that above, we can obtain a $(15, 15, 5)$–PF. The polynomials $g(X)$ can be taken to be $(X - 1)^2(X^4 + X^3 + X^2 + X + 1)^2$ in $Z_3[X]$ and $(X - 1)^2(X^2 + X + 1)^4$ in $Z_5[X]$.

7.2.5. Perfect Factors for $v = 6$. By Lemma 7.1, only the following six parameter sets need to be considered:

$$(10, 10, 6), (12, 6, 6), (15, 15, 6), (20, 10, 6), (30, 30, 6), \text{ and } (60, 30, 6).$$

PFs with parameters $(12, 6, 6)$, $(20, 10, 6)$, and $(60, 30, 6)$ can be obtained by applying Construction 4.1 with $t = 2$ to PFs with parameters $(6, 6, 3)$, $(10, 10, 3)$, and $(30, 30, 3)$, respectively (c.f. section 3.4 of [9]).

Consider the parameters $(10, 10, 6)$. A $(5, 2, 5, 6)$–GPF can be obtained using the polynomial $(X - 1)(X + 1)^3$ in $Z_5[X]$. We can obtain a $(2, 5, 2, 6)$-PMF using Construction 2.8 by taking $g(X) = X^4 + X^3 + X^2 + X + 1$ and $b(X) = 1$ in $Z_2[X]$. Combining these using Theorem 1.24, we obtain a $(10, 10, 6)$–PF.

A $(15, 15, 6)$–PF can be obtained by combining GPFs constructed using the polynomials $(X - 1)(X^4 + X^3 + X^2 + X + 1)^2$ in $Z_3[X]$ and $(X - 1)(X^2 + X + 1)^4$ in $Z_5[X]$.

Finally, consider the parameters $(30, 30, 6)$. It is easy to see from cyclotomic factorizations how to obtain degree 24 factors $g(X)$ of $X^{30} - 1$ in each of $Z_3[X]$ and $Z_5[X]$. These can be used to construct a $(3, 10, 3, 6)$–GPF and a $(5, 6, 5, 6)$–GPF. Combining these using Construction 1.13, by Theorem 1.22 we obtain a $(15, 2, 15, 6)$–GPF. By Theorem 1.12, there exists a $(2, 15, 2, 6)$–PMF. Applying Theorem 1.24, we can obtain a $(30, 30, 6)$–PF.

7.3. Perfect Factors for $v = 7$ and $v = 8$. We finally consider the existence of perfect factors for $v = 7$ and $v = 8$, and in doing so list the smallest undecided cases.

By Lemma 7.1, for $v = 7$, the following 17 parameter sets need to be considered:

$$\begin{array}{cccccc} (10, 10, 7), & (12, 6, 7), & (14, 14, 7), & (15, 15, 7), & (20, 10, 7), & (21, 21, 7), \\ (28, 14, 7), & (30, 30, 7), & (35, 35, 7), & (42, 42, 7), & (60, 30, 7), & (70, 70, 7), \\ (84, 42, 7), & (105, 105, 7), & (140, 70, 7), & (210, 210, 7), & \text{and} & (420, 210, 7). \end{array}$$

All these parameter sets, except $(10, 10, 7)$, $(12, 6, 7)$, $(15, 15, 7)$, $(20, 10, 7)$, $(30, 30, 7)$, $(35, 35, 7)$, and $(60, 60, 7)$, fall to Theorem 2.7. Constructions based on cyclic codes can be used to build PFs for 6 out of these 7 remaining sets (we omit the details), the parameters $(10, 10, 7)$ resisting attack by such methods.

Similarly when $v = 8$, 14 of the 24 parameter sets that remain after applying Lemma 7.1 fall to Construction 4.1 with $t = 2$. All but one of the remaining ten sets then fall to constructions based on cyclic codes. The parameter set $(10, 10, 8)$ remains undecided.

One reason for the difficulty with the sets $(10, 10, 7)$ and $(10, 10, 8)$ is that $X^{10} - 1$ has no factors of degrees 2 or 3 in $Z_2[X]$ that are suitable for use in our cyclic code constructions. If a $(10, 10, 7)$–PF and a $(10, 10, 8)$–PF could be shown to exist, then Conjecture 1.4 would also be true for $v \leq 8$. Such PFs would contain 10^6 and 10^7 cycles of period 10, respectively, and as such appear to be out of the reach of computer search.

8. Conclusions. We have provided further evidence to support the conjecture that the necessary conditions of Lemma 1.3 are sufficient for the existence of a Perfect Factor. Indeed it is probably possible to extend our case by case analysis to cover most parameter sets for $v = 9$ and beyond.

More importantly, we have provided new and powerful construction methods which may have the potential to help establish the conjecture for general v. In this direction it may be worthwhile examining in more detail the different ways in which these methods can be combined to produce Perfect Factors. We have already done this for interleaving combined with the Lempel inverse homomorphism in section 6 of this paper.

It is also worth noting that we have only used the coding-theoretic methods developed here to attack the existence question for small v. However, even for small v, these methods do have some limitations, as illustrated by our failure with parameters $(10, 10, 7)$ and $(10, 10, 8)$. Indeed, it is not hard to show that if $p \geq 5$ is prime and 2 is primitive modulo p, then $X^{2p} - 1$ has factorization $(X+1)^2(X^{p-1} + X^{p-2} + \cdots + 1)^2$ in $\mathrm{Z}_2[X]$. So, in this case, $X^{2p} - 1$ has no factors of degrees $2, 3, \cdots, p-2$ that can be used in our cyclic code constructions. This means that the cyclic code techniques in this paper cannot be used to help construct $(2p, 2p, v)$–PFs for any v with $p+2 \leq v \leq 2p-2$. These are examples of parameter sets for which no construction methods are currently known.

Acknowledgment. We would like to thank an anonymous referee for valuable comments.

REFERENCES

[1] T. Etzion, *Constructions for perfect maps and pseudo-random arrays*, IEEE Trans. Inform. Theory, 34 (1988), pp. 1308–1316.
[2] H. Fredricksen, *A survey of full length nonlinear shift register cycle algorithms*, SIAM Rev., 24 (1982), pp. 195–221.
[3] G. Hurlbert and G. Isaak, *On the de Bruijn torus problem*, J. Combin. Theory Ser. A, 64 (1993), pp. 50–62.
[4] A. Lempel, *On a homomorphism of the de Bruijn graph and its application to the design of feedback shift registers*, IEEE Trans. Comput., C-19 (1970), pp. 1204–1209.
[5] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, Cambridge, 1986.
[6] F. MacWilliams and N. Sloane, *The theory of error-correcting codes*, North-Holland, Amsterdam, 1977.
[7] C. Mitchell, *Constructing c-ary perfect factors*, Des. Codes Cryptogr., 4 (1994), pp. 341–368.
[8] C. Mitchell, *New c-ary perfect factors in the de Bruijn graph*, in Codes and Cyphers, in Proceedings of the 4th IMA Conference on Cryptography and Coding, Cirencester, December 1993, P. Farrell, ed., Formara Ltd., Southend, 1995, pp. 299–313.

[9] C. Mitchell, *De Bruijn sequences and perfect factors*, SIAM J. Discrete Math., 10 (1997), pp. 270–281.
[10] C. Mitchell and K. Paterson, *Decoding perfect maps*, Des. Codes Cryptogr., 4 (1994), pp. 11–30.
[11] K. Paterson, *Perfect maps*, IEEE Trans. on Inform. Theory, 40 (1994), pp. 743–753.
[12] K. Paterson, *Perfect factors in the de Bruijn graph*, Des. Codes Cryptogr., 5 (1995), pp. 115–138.
[13] K. Paterson, *New classes of perfect maps* I, J. Combin. Theory Ser. A, 73 (1996), pp. 302–334.
[14] K. Paterson, *New classes of perfect maps* II, J. Combin. Theory Ser. A, 73 (1996), pp. 335–345.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 265–272, May 1998

006

ISOMORPHISM CLASSES OF CONCRETE GRAPH COVERINGS*

RONGQUAN FENG†, JIN HO KWAK‡, JUYOUNG KIM§, AND JAEUN LEE¶

Abstract. Hofmeister introduced the notion of a concrete (resp., concrete regular) covering of a graph G and gave formulas for enumerating the isomorphism classes of concrete (resp., concrete regular) coverings of G [*Ars Combin.*, 32 (1991), pp. 121–127; *SIAM J. Discrete Math.*, 8 (1995), pp. 51–61]. In this paper, we show that the number of the isomorphism classes of n-fold concrete (resp., concrete regular) coverings of G is equal to that of the isomorphism classes of n-fold (resp., regular) coverings of a new graph, the join $G+\infty$ of G and an extra vertex ∞. As a consequence, we can enumerate the isomorphism classes of concrete (resp., concrete regular) coverings of a graph by using known formulas for enumerating the isomorphism classes of coverings (resp., regular coverings) of a graph.

Key words. concrete graph coverings, voltage assignments, enumeration

AMS subject classifications. 05C10, 05C30, 57M15

PII. S089548019630443X

1. Introduction.

Let G be a connected finite simple graph with vertex set $V(G)$ and edge set $E(G)$. The *neighborhood* of a vertex $v \in V(G)$, denoted by $N(v)$, is the set of vertices adjacent to v. We use $|X|$ for the cardinality of a set X. An *automorphism* of G is a permutation of the vertex set $V(G)$ which preserves adjacency. Obviously, a composition of two automorphisms is also an automorphism. Hence the automorphisms of G form a permutation group, $\mathrm{Aut}\,(G)$, which acts on the vertex set $V(G)$.

A graph $\widetilde{G}$ is called a *covering* of G with projection $p : \widetilde{G} \to G$ if there is a surjection $p : V(\widetilde{G}) \to V(G)$ such that $p|_{N(\tilde{v})} : N(\tilde{v}) \to N(v)$ is a bijection for all vertex $v \in V(G)$ and $\tilde{v} \in p^{-1}(v)$. We also say that the projection $p : \widetilde{G} \to G$ is an *n-fold covering* of G if p is n-to-one. A covering $p : \widetilde{G} \to G$ is said to be *regular* if there is a subgroup $\mathcal{A}$ of the automorphism group $\mathrm{Aut}(\widetilde{G})$ of $\widetilde{G}$ acting freely on $\widetilde{G}$ such that the quotient graph $\widetilde{G}/\mathcal{A}$ is isomorphic to G. In fact, the group $\mathcal{A}$ is the covering transformation group of the covering $p : \widetilde{G} \to G$. The fiber of an edge or a vertex is its preimage under p.

An n-fold covering $p : \widetilde{G} \to G$ is said to be *concrete* if it is accompanied by an explicit partition $\mathcal{P} = \{P_1, \ldots, P_n\}$ of $V(\widetilde{G})$ such that every partition set P_i meets every vertex fiber exactly once; we write $(p, \mathcal{P})$ for short. The partition sets P_i are the *sheets* of p. A *concrete regular* covering is a concrete covering $(p, \mathcal{P})$, in which $p : \widetilde{G} \to G$ is regular and every covering transformation of $\widetilde{G}$ preserves the sheets in $\mathcal{P}$.

*Received by the editors May 28, 1996; accepted for publication (in revised form) March 10, 1997. The research of R. Feng and J. H. Kwak was supported by a grant from POSTECH/BSRI-Special Fund.

http://www.siam.org/journals/sidma/11-2/30443.html

†Department of Mathematics, Peking University, Beijing 100 871, P.R. China (fengrq@sxx0.math.pku.edu.cn).

‡Department of Mathematics, Pohang University of Science and Technology, Pohang 790-784, Korea (jinkwak@postech.ac.kr).

§Department of Mathematics, Catholic University of Taegu-Hyosung, Kyongsan 713-702, Korea (jykim@cuth.cataegu.ac.kr).

¶Department of Mathematics, Yeungnam University, Kyongsan 712-749, Korea (julee@ynucc.yeungnam.ac.kr). The research of this author was supported by TGRC-KOSEF.

Let Γ be a group of automorphisms of the graph G. Two coverings $p_i : \widetilde{G}_i \to G$, $i = 1, 2$ are said to be *isomorphic with respect to* Γ if there exist a graph isomorphism $\Phi : \widetilde{G}_1 \to \widetilde{G}_2$ and a graph automorphism $\gamma \in \Gamma$ such that the diagram

$$\begin{array}{ccc} \widetilde{G}_1 & \xrightarrow{\Phi} & \widetilde{G}_2 \\ {\scriptstyle p_1}\downarrow & & \downarrow{\scriptstyle p_2} \\ G & \xrightarrow{\gamma} & G \end{array}$$

commutes. Such a Φ is called a *covering isomorphism with respect to* Γ. Two concrete coverings $(p_i, \mathcal{P}_i), i = 1, 2$ are said to be *isomorphic with respect to* Γ if p_1 and p_2 are isomorphic in the sense of the above commutative diagram with a sheet preserving map Φ. Note that for any group Γ of automorphisms of G, the (concrete) covering isomorphic relation with respect to Γ on the (concrete) coverings of G is an equivalence relation.

Every edge of a graph G gives rise to a pair of oppositely directed edges. By $e^{-1} = vu$, we mean the reverse edge to a directed edge $e = uv$. We denote the set of directed edges of G by $D(G)$. Following Gross and Tucker [2], [3], a *permutation voltage assignment* ϕ of G is a function $\phi : D(G) \to S_n$ with the property that $\phi(e^{-1}) = \phi(e)^{-1}$ for each $e \in D(G)$, where S_n is the symmetric group on n elements $\{1, \ldots, n\}$. The *permutation derived graph* G^ϕ is defined as follows: $V(G^\phi) = V(G) \times \{1, \ldots, n\}$, and for each edge $e = uv \in D(G)$ and $j \in \{1, \ldots, n\}$ let there be an edge (e, j) in $D(G^\phi)$ joining a vertex (u, j) and $(v, \phi(e)j)$. The first coordinate projection $p^\phi : G^\phi \to G$, called the natural projection, is an n-fold covering. Let $\mathcal{A}$ be a finite group. An *ordinary voltage assignment* (or, $\mathcal{A}$-*voltage assignment*) of G is a function $\phi : D(G) \to \mathcal{A}$ with the property that $\phi(e^{-1}) = \phi(e)^{-1}$ for each $e \in D(G)$. The values of ϕ are called *voltages*, and $\mathcal{A}$ is called the *voltage group*. The *ordinary derived graph* $G \times_\phi \mathcal{A}$ derived from an ordinary voltage assignment $\phi : D(G) \to \mathcal{A}$ has as its vertex set $V(G) \times \mathcal{A}$ and as its edge set $E(G) \times \mathcal{A}$ so that an edge of $G \times_\phi \mathcal{A}$ joins a vertex (u, g) to $(v, \phi(e)g)$ for $e = uv \in D(G)$ and $g \in \mathcal{A}$. In the (ordinary) derived graph $G \times_\phi \mathcal{A}$, a vertex (u, g) is denoted by u_g, and an edge (e, g) by e_g. The first coordinate projection $p_\phi : G \times_\phi \mathcal{A} \to G$, called the natural projection, commutes with the left multiplication action of the $\phi(e)$ and the right action of $\mathcal{A}$ on the fibers, which is free and transitive, so that p_ϕ is an $|\mathcal{A}|$-fold regular covering, called simply an $\mathcal{A}$-*covering*.

Let $C^1(G; n)$ (resp., $C^1(G; \mathcal{A})$) denote the set of all permutation voltage assignments $\phi : D(G) \to S_n$ (resp., $\mathcal{A}$-voltage assignments $\phi : D(G) \to \mathcal{A}$) of G. For a spanning tree T of G, let $C^1_T(G; n)$ (resp., $C^1_T(G; \mathcal{A})$) denote the set of elements ϕ of $C^1(G; n)$ (resp., $C^1(G; \mathcal{A})$) such that $\phi(uv) =$ identity for each $uv \in D(T)$. Gross and Tucker [2], [3] showed that every n-fold covering (resp., regular covering) $\widetilde{G}$ of a graph G can be derived from a permutation (resp., ordinary) voltage assignment in $C^1_T(G; n)$ (resp., $C^1_T(G; \mathcal{A})$ for a group $\mathcal{A}$ of order n). Hofmeister [8], [9] proved that every concrete covering (resp., concrete regular covering) $\widetilde{G}$ of a graph G can be derived from a permutation (resp., ordinary) voltage assignment with sheets $P_i = \{(v, i) | v \in V(G)\}$, $i = 1, \ldots, n$ (resp., $P_a = \{(v, a) | v \in V(G)\}$, $a \in \mathcal{A}$).

In this paper, we show that the number of the isomorphism classes of n-fold concrete (resp., concrete regular) coverings of G is equal to that of the isomorphism classes of n-fold (resp., regular) coverings of a new graph, the join $G + \infty$ of G and

an extra vertex ∞. This means that we can enumerate the isomorphism classes of concrete (resp., concrete regular) coverings of a graph by using known formulas for enumerating the isomorphism classes of coverings (resp., regular coverings) of a graph, which can be found in [6], [7], [10], [11], [15], [17], and [18].

2. Coverings and concrete coverings. In this section, we show that there exists a one-to-one correspondence between the isomorphism classes of concrete (resp., regular) coverings of a graph G and the isomorphism classes of (resp., regular) coverings of a new graph G_∞, the join $G + \infty$ of G and an extra vertex ∞.

For a group Γ of automorphisms of a graph G, let $\mathrm{Iso}_\Gamma(G;n)$ (resp., $\mathrm{Iso}_\Gamma^R(G;n)$, $\mathrm{Iso}_\Gamma^C(G;n)$, $\mathrm{Iso}_\Gamma^{CR}(G;n)$) denote the number of the isomorphism classes of n-fold (resp., regular, concrete, concrete regular) coverings of G with respect to Γ, and let $\mathrm{Iso}_\Gamma(G;\mathcal{A})$ (resp., $\mathrm{Iso}_\Gamma^C(G;\mathcal{A})$) denote the number of the isomorphism classes of (resp., concrete) $\mathcal{A}$-coverings of G with respect to Γ.

Let G be a graph and let $G_\infty = G + \infty$ be the join of G and an extra vertex ∞, i.e., G_∞ consists of a copy of G and ∞ with additional edges joining every vertex of G to the vertex ∞. It is clear that an automorphism of G can be uniquely extended to an automorphism of G_∞ which fixes ∞. Hence, from now on we can identify a group of automorphisms of G with the group of corresponding automorphisms of G_∞. Let T_∞ be the spanning tree of G_∞ with $E(T_\infty) = \{\infty v \mid v \in V(G)\}$. Then $\gamma(T_\infty) = T_\infty$ for every automorphism γ of G because $\gamma(\infty) = \infty$.

THEOREM 2.1. *Let G be a graph and Γ a group of automorphisms of G. Then*

$$\mathrm{Iso}_\Gamma^C(G;n) = \mathrm{Iso}_\Gamma(G_\infty;n)$$

for any natural number n.

Proof. It is known [8] that any two n-fold concrete coverings G^ϕ and G^ψ are isomorphic with respect to Γ as concrete coverings if and only if there exist a permutation $\sigma \in S_n$ and an automorphism $\gamma \in \Gamma$ such that $\psi(\gamma(u)\gamma(v)) = \sigma\phi(uv)\sigma^{-1}$ for each $uv \in D(G)$. It is also known [14] that for an n-fold covering $p : \widetilde{G_\infty} \to G_\infty$, there exists a voltage assignment $\phi^* \in C^1_{T_\infty}(G_\infty;n)$ such that the covering $p^{\phi^*} : G_\infty^{\phi^*} \to G_\infty$ is isomorphic to the covering $p : \widetilde{G_\infty} \to G_\infty$ with respect to the trivial automorphism group $\{1\}$ of G_∞, and two coverings $p^{\phi^*} : G_\infty^{\phi^*} \to G_\infty$ and $p^{\psi^*} : G_\infty^{\psi^*} \to G_\infty$, $\phi^*, \psi^* \in C^1_{T_\infty}(G_\infty;n)$ are isomorphic with respect to Γ if and only if there exist a permutation $\sigma \in S_n$ and an automorphism $\gamma \in \Gamma$ such that $\psi^*(\gamma(u)\gamma(v)) = \sigma\phi^*(uv)\sigma^{-1}$ for each $uv \in D(G_\infty) - D(T_\infty) = D(G)$.

Define $f : C^1_{T_\infty}(G_\infty;n) \to C^1(G;n)$ by $f(\phi^*) = \phi^*|_G$, the restriction of ϕ^* on G. Then f is bijective and, by the above discussion, two coverings $p^{\phi^*} : G_\infty^{\phi^*} \to G_\infty$ and $p^{\psi^*} : G_\infty^{\psi^*} \to G_\infty$ are isomorphic with respect to Γ if and only if the two concrete coverings $p^{f(\phi^*)} : G^{f(\phi^*)} \to G$ and $p^{f(\psi^*)} : G^{f(\psi^*)} \to G$ are isomorphic with respect to Γ. This completes the proof. □

In a way similar to Theorem 3 in [12], we can prove the following theorem.

THEOREM 2.2. *Let G be a graph and Γ a group of automorphisms of G. Then*

$$\mathrm{Iso}_\Gamma^C(G;\mathcal{A}) = \mathrm{Iso}_\Gamma(G_\infty;\mathcal{A})$$

for any finite group $\mathcal{A}$ and

$$\mathrm{Iso}_\Gamma^{CR}(G;n) = \mathrm{Iso}_\Gamma^R(G_\infty;n)$$

for any natural number n.

3. Enumeration of concrete coverings. In this section, we obtain an enumeration formula for the isomorphism classes of concrete coverings of a graph G as a reformulation of Hofmeister's enumeration formula for those of concrete coverings.

To do this, we recall some notations in [14]. Let T be a spanning tree in a graph H. For an automorphism γ of H with $\gamma(T) = T$, we define an equivalence relation $\sim_\gamma$ on $D(H) - D(T)$ as follows: $e_1 \sim_\gamma e_2$ if and only if $e_1 = \gamma^\ell e_2$ for some ℓ. An equivalence class $[e]$ of e is called *of class* 1 if both e and e^{-1} are contained in the same equivalence class and is called *of class* 2 otherwise. For any edge $e \in D(H) - D(T)$, we define a number $\eta(\gamma, e)$ to be the smallest natural number ℓ such that $e^{-1} = \gamma^\ell e$ if $[e]$ is of class 1, and the smallest natural number ℓ such that $e = \gamma^\ell e$ if $[e]$ is of class 2. This number is well defined because γ has finite order in Γ. Note that if $H = G_\infty$ and $T = T_\infty$, then $D(G_\infty) - D(T_\infty) = D(G)$. Recall that $\mathrm{Aut}\,(G)$ can be regarded as the set of all automorphisms γ of G_∞ such that $\gamma(T_\infty) = T_\infty$.

Now, we can deduce the following from Theorem 2.1 and Theorem 3 in [14].

THEOREM 3.1. *For any subgroup* Γ *of* $\mathrm{Aut}\,(G)$,

$$\begin{aligned}\mathrm{Iso}^C_\Gamma(G;n) &= \mathrm{Iso}_\Gamma(G_\infty;n)\\ &= \frac{1}{|\Gamma|n!}\sum_{(\gamma,\sigma)\in\Gamma\times S_n}\left(\prod_{[e]\in\text{Class 1}}|I(\sigma^{\eta(\gamma,e)})|\right)\left(\prod_{[e]\in\text{Class 2}}|F(\sigma^{\eta(\gamma,e)})|\right)^{\frac{1}{2}},\end{aligned}$$

where the product over the empty index set is defined as 1 *and, for any natural number* r, $I(\sigma^r) = \{\tau \in S_n : \sigma^r\tau\sigma^{-r} = \tau^{-1}\}$ *and* $F(\sigma^r) = \{\tau \in S_n : \sigma^r\tau\sigma^{-r} = \tau\}$ *are defined as subsets of the symmetric group* S_n.

For a permutation $\sigma \in S_n$, let $(\ell_1, \ldots, \ell_n)$ be the cycle type of σ, i.e., for each integer i from 1 to n, a decomposition of σ into a product of disjoint cycles has exactly ℓ_i cycles of length i. Then $\ell_1 + 2\ell_2 + \cdots + n\ell_n = n$ and

$$|F(\sigma)| = \ell_1!2^{\ell_2}\ell_2!, \cdots, n^{\ell_n}\ell_n!.$$

To compute $|I(\sigma)|$, let $\nu = \tau\sigma$ in S_n, then $\sigma\tau\sigma^{-1} = \tau^{-1}$ if and only if $\nu^2 = \sigma^2$, so the number $|I(\sigma)|$ is equal to the number of ν in S_n such that $\nu^2 = \sigma^2$. By using this, we can see that

$$|I(\sigma)| = \sum_{\substack{t_i = k_i,\ i \text{ odd and } 2i > n,\\ t_{2i} = \frac{1}{2}k_i,\ i \text{ even and } 2i \le n,\\ t_i + 2t_{2i} = k_i,\ i \text{ odd and } 2i \le n}} \frac{k_1!2^{k_2}k_2!, \cdots, n^{k_n}k_n!}{t_1!2^{t_2}t_2!, \cdots, n^{t_n}t_n!},$$

where $(k_1, \ldots, k_n)$ is the cycle type of σ^2, i.e.,

$$k_i = \begin{cases} \ell_i & \text{if } i \text{ is odd and } 2i > n,\\ 0 & \text{if } i \text{ is even and } 2i > n,\\ \ell_i + 2\ell_{2i} & \text{if } i \text{ is odd and } 2i \le n,\\ 2\ell_{2i} & \text{if } i \text{ is even and } 2i \le n.\end{cases}$$

Let $\gamma \in \mathrm{Aut}(G)$ and $|V(G)| = m$; then γ is a permutation on m elements, i.e., $\gamma \in S_m$ (γ also can be considered as a permutation on $V(G_\infty)$ satisfying $\gamma(\infty) = \infty$). Also, γ can be decomposed into a product of disjoint cycles. If γ has no even length cycle, then every equivalence class of edges in $D(G)(= D(G_\infty) - D(T_\infty))$ is of class 2. If γ has an even length cycle, say $(u_1u_2, \cdots, u_{2k})$ is such a cycle, and $u_tu_{k+t} \in D(G)$

for some t, then $u_t u_{k+t} \in D(G)$ for every t, $[e] = \{\, u_i u_j \mid |i-j| = k \,\}$ is of class 1 and $\eta(\gamma, e) = k$. Conversely, every equivalence class of class 1 can be obtained in this way. Now suppose that $[e]$ is of class 2; let $e = uv$ be an element of $[e]$. If u, v are in the same cycle of γ, then $\eta(\gamma, e)$ is the length of this cycle. If u, v are in different cycles of γ, then $\eta(\gamma, e)$ is the least common multiple of the lengths of these two cycles containing u and v, respectively. In this way, we can calculate $\mathrm{Iso}_\Gamma^C(G; n) = \mathrm{Iso}_\Gamma(G_\infty; n)$ for any graph G and any $\Gamma \leq \mathrm{Aut}(G)$. For example, if γ is the identity, then every equivalence class of $e \in D(G)$ is of class 2 and contains only one edge and $\eta(\gamma, e) = 1$, i.e., there are $2|E(G)|$ equivalence classes of class 2. Therefore, we have the following corollary.

COROLLARY 3.2. *The number* $\mathrm{Iso}_{\{1\}}^C(G; n)$ *of the isomorphism classes of* n*-fold concrete coverings of a graph* G *with respect to the trivial automorphism group* $\{1\}$ *is*

$$\mathrm{Iso}_{\{1\}}^C(G; n) = \sum_{\ell_1 + 2\ell_2 + \cdots + n\ell_n = n} (\ell_1!\, 2^{\ell_2}\ell_2!, \cdots, n^{\ell_n}\ell_n!)^{|E(G)|-1}.$$

4. Enumeration of concrete regular coverings. Hofmeister [9] gave a general enumeration formula for the isomorphism classes of concrete regular coverings with respect to a group of automorphisms of a given base graph. In this section, we obtain new enumeration formulas for the isomorphism classes of concrete regular coverings as a reformulation of Hofmeister's enumeration formulas for those of concrete regular coverings.

For convenience, we denote by $\mathrm{Isoc}_\Gamma^R(G; n)$ (resp., $\mathrm{Isoc}_\Gamma(G; \mathcal{A})$) the number of the isomorphism classes of connected regular (resp., connected $\mathcal{A}$-) coverings of G with respect to a group Γ of automorphisms of G which fixes a spanning tree T of G.

Now, the following theorem comes from our Theorem 2.2 and from Theorems 4, 5, and 6 in [13].

THEOREM 4.1. *Let* G *be a graph and* Γ *a group of automorphisms of* G*. Let* $\mathcal{A}$ *be a finite group. Then*

$$\mathrm{Iso}_\Gamma^C(G; \mathcal{A}) = \mathrm{Iso}_\Gamma(G_\infty; \mathcal{A}) = \sum_{\mathcal{S}} \mathrm{Isoc}_\Gamma(G_\infty; \mathcal{S}),$$

where $\mathcal{S}$ *runs over all of the isomorphism classes of subgroups of* $\mathcal{A}$*,*

$$\mathrm{Iso}_\Gamma^{CR}(G; n) = \mathrm{Iso}_\Gamma^R(G_\infty; n) = \sum_{d|n} \mathrm{Isoc}_\Gamma^R(G_\infty; d),$$

and

$$\mathrm{Isoc}_\Gamma^R(G_\infty; d) = \sum_{\mathcal{B}} \mathrm{Isoc}_\Gamma(G_\infty; \mathcal{B}),$$

where $\mathcal{B}$ *runs over all of the isomorphism classes of groups of order* d*.*

A finite group $\mathcal{A}$ is said to have the *isomorphism extension property* if every isomorphism between any two isomorphic subgroups of $\mathcal{A}$ can be extended to an automorphism of $\mathcal{A}$. We divide $D(G)$ into equivalence classes of classes 1 and 2 for each $\gamma \in \Gamma$ in the same way as in section 3. From our Theorem 2.2 and from Theorem 5 in [12], we can deduce the following.

THEOREM 4.2. *Let* G *be a graph and* Γ *a group of automorphisms of* G*. Let* $\mathcal{A}$ *be a finite group with the isomorphism extension property. Then*

$$\mathrm{Iso}_\Gamma^C(G;\mathcal{A}) = \mathrm{Iso}_\Gamma(G_\infty;\mathcal{A})$$
$$= \frac{1}{|\Gamma||\mathrm{Aut}(\mathcal{A})|}\sum_{(\gamma,\sigma)\in\Gamma\times\mathrm{Aut}(\mathcal{A})}\left(\prod_{[e]\in\text{Class 1}}|I(\sigma^{\eta(\gamma,e)})|\right)\left(\prod_{[e]\in\text{Class 2}}|F(\sigma^{\eta(\gamma,e)})|\right)^{\frac{1}{2}},$$

where the product over the empty index set is defined as 1 *and, for any natural number* r, $I(\sigma^r) = \{g \in \mathcal{A} : \sigma^r(g) = g^{-1}\}$ *and* $F(\sigma^r) = \{g \in \mathcal{A} : \sigma^r(g) = g\}$ *are defined as subsets of* $\mathcal{A}$.

For the trivial automorphism group $\{1\}$ of G, we have the following corollary.

COROLLARY 4.3. *If a finite group* $\mathcal{A}$ *has the isomorphism extension property, then the number* $\mathrm{Iso}_{\{1\}}^C(G;\mathcal{A})$ *of the isomorphism classes of concrete* $\mathcal{A}$*-coverings of a graph* G *with respect to the trivial automorphism group* $\{1\}$ *is*

$$\mathrm{Iso}_{\{1\}}^C(G;\mathcal{A}) = \frac{1}{|\mathrm{Aut}(\mathcal{A})|}\sum_{\sigma\in\mathrm{Aut}(\mathcal{A})}|F(\sigma)|^{|E(G)|},$$

where $F(\sigma) = \{g \in \mathcal{A} : \sigma(g) = g\}$.

Kwak, Chun, and Lee [13] obtained the following.

THEOREM 4.4. *Let* G *be a graph and* $\mathcal{A}$ *a finite group. Then*

$$\mathrm{Isoc}_{\{1\}}(G_\infty;\mathcal{A}) = \frac{|\mathfrak{G}(\mathcal{A};|E(G)|)|}{|\mathrm{Aut}(\mathcal{A})|},$$

where $\mathfrak{G}(\mathcal{A};n) = \{(g_1,g_2,\ldots,g_n) \in \mathcal{A}^n : \{g_1,g_2,\ldots,g_n\}$ *generates* $\mathcal{A}\}$.

In [13], Kwak, Chun, and Lee calculated the number $\mathrm{Isoc}_{\{1\}}(G;\mathcal{A})$ of the isomorphism classes of connected $\mathcal{A}$-coverings of G for any finite abelian group $\mathcal{A}$. By using their results in [13] and our Theorem 2.2, we have the following.

COROLLARY 4.5. *Let* G *be a graph,* p *a prime, and* $m > 0$. *Then*

$$\mathrm{Iso}_{\{1\}}^C(G;\mathbb{Z}_{p^m}) = \begin{cases} 1+m, & \text{if } G = K_2, \\ 1+\dfrac{p^{|E(G)|}-1}{p-1}\cdot\dfrac{p^{m(|E(G)|-1)}-1}{p^{|E(G)|-1}-1}, & \text{otherwise,}\end{cases}$$

and

$$\mathrm{Iso}_{\{1\}}^C(G;m\mathbb{Z}_p) = 1+\sum_{h=1}^{m}\frac{(p^{|E(G)|}-1)(p^{|E(G)|-1}-1)\cdots(p^{|E(G)|-h+1}-1)}{(p^h-1)(p^{h-1}-1)\cdots(p-1)},$$

where $\mathbb{Z}_{p^m}$ *is the cyclic group of order* p^m *and* $m\mathbb{Z}_p$ *is the direct sum of* m *copies of* $\mathbb{Z}_p$.

5. Applications. In this section, we give a one-to-one correspondence between the isomorphism classes of concrete double coverings of a graph G with respect to a group Γ of automorphisms of G and the isomorphism classes of spanning subgraphs of G with respect to Γ, where, by definition, two subgraphs S and H are isomorphic with respect to Γ if there exists a $\gamma \in \Gamma$ such that $\gamma(H) = S$.

Let $\mathbb{Z}_2 = \{0,1\}$ be the additive group of order 2. Then every $\mathbb{Z}_2$-voltage assignment ϕ in $C^1(G;\mathbb{Z}_2)$ can be regarded as a map $\phi : E(G) \to \mathbb{Z}_2$ because the inverse of any element g of $\mathbb{Z}_2$ is g itself. The subgraph $G[\phi]$ of G *associated with* ϕ is defined as

$$V(G[\phi]) = V(G), \qquad E(G[\phi]) = \{\{u,v\}|\phi(\{u,v\}) = 1\}.$$

Then $G[\phi]$ is a spanning subgraph of G and every spanning subgraph of G can be obtained in this way. Moreover, it gives a one-to-one correspondence between the set of all spanning subgraphs of G and the set $C^1(G;\mathbb{Z}_2)$ of all $\mathbb{Z}_2$-voltage assignments of G.

It is easy to show that for any two voltage assignments ϕ and ψ in $C^1(G;\mathbb{Z}_2)$, $G[\phi]$ and $G[\psi]$ are isomorphic with respect to a group Γ of automorphisms of G if and only if there exists a $\gamma \in \Gamma$ such that $\phi(\{u,v\}) = \psi(\{\gamma(u),\gamma(v)\})$ for all $\{u,v\} \in E(G)$.

As in the proof of Theorem 2.1, we can see that two concrete double coverings of G, $G \times_\phi \mathbb{Z}_2$, and $G \times_\psi \mathbb{Z}_2$, derived from ϕ and ψ, respectively, are isomorphic with respect to Γ if and only if $\phi(\{u,v\}) = \psi(\{\gamma(u),\gamma(v)\})$ for all $\{u,v\} \in E(G)$.

Now, we summarize our discussions as follows.

THEOREM 5.1. *Let G be a graph and Γ a group of automorphisms of G. Then the number $\mathrm{Iso}^C_\Gamma(G;2)$ of the isomorphism classes of concrete double coverings of G with respect to Γ is equal to the number of the isomorphism classes of spanning subgraphs of G with respect to Γ.*

Notice that the number of the isomorphism classes of spanning subgraphs of G with respect to $\mathrm{Aut}\,(G)$ was already estimated in [5]. In fact, in this paper we give a new method to estimate that number by giving a formula for counting the number $\mathrm{Iso}^C_\Gamma(G;2)$.

Since the set of all spanning subgraphs of the complete graph K_n on n vertices is just the set of all graphs with n vertices and $\mathrm{Aut}\,(K_n)$ is the symmetric group S_n, we have the following.

COROLLARY 5.2. *The number $\mathrm{Iso}^C_{S_n}(K_n;2)$ of the isomorphism classes of concrete double coverings of K_n with respect to $\mathrm{Aut}\,(K_n) = S_n$ is equal to the number g_n of the isomorphism classes of graphs with n vertices.*

The following comes from Theorem 4.2.

THEOREM 5.3. *Let G be a graph and Γ a group of automorphisms of G. Then the number $\mathrm{Iso}^C_\Gamma(G;2)$ of the isomorphism classes of concrete double coverings of G with respect to Γ is*

$$\mathrm{Iso}^C_\Gamma(G;2) = \mathrm{Iso}_\Gamma(G_\infty;2) = \frac{1}{|\Gamma|}\sum_{\gamma\in\Gamma} 2^{|E(G/\langle\gamma\rangle)|} = \sum_{\gamma} \frac{2^{|E(G/\langle\gamma\rangle)|}}{|Z(\gamma)|},$$

where γ in the latter summation runs over all representatives of the conjugacy classes of Γ, $G/\langle\gamma\rangle$ is the quotient graph induced by the action of the subgroup $\langle\gamma\rangle$ generated by γ, and $Z(\gamma)$ is the center of γ in Γ.

If $G = K_n$, then for each $\gamma \in \mathrm{Aut}\,(K_n) = S_n$ with cycle type $(\ell_1,\ell_2,\ldots,\ell_n)$,

$$|Z(\gamma)| = \ell_1!2^{\ell_2}\ell_2!,\cdots,n^{\ell_n}\ell_n! \quad \text{and} \quad |E(K_n/\langle\gamma\rangle)| = \frac{1}{2}\left(\sum_{s,t=1}^{n}\ell_s\ell_t(s,t) - \sum_{s=\mathrm{odd}}\ell_s\right),$$

where (s,t) denotes the greatest common divisor of s and t. Now, the following comes from these facts and Theorem 5.3.

COROLLARY 5.4. *The number g_n of the isomorphism classes of graphs with n vertices is*

$$g_n = \mathrm{Iso}^C_{S_n}(K_n;2) = \sum_{\ell_1+2\ell_2+\cdots+n\ell_n=n} \frac{2^{N(\ell_1,\ell_2,\ldots,\ell_n)}}{\ell_1!2^{\ell_2}\ell_2!,\cdots,n^{\ell_n}\ell_n!},$$

where

$$N(\ell_1, \ell_2, \ldots, \ell_n) = \frac{1}{2}\left(\sum_{s,t=1}^{n} \ell_s \ell_t (s,t) - \sum_{s=odd} \ell_s\right)$$

and (s,t) *denotes the greatest common divisor of* s *and* t.

In fact, the number g_n was already known in [1] and [4].

REFERENCES

[1] R. L. Davis, *The number of structures of finite relations*, Proc. Amer. Math. Soc., 4 (1953), pp. 486–495.

[2] J. L. Gross and T. W. Tucker, *Generating all graph coverings by permutation voltage assignments*, Discrete Math., 18 (1977), pp. 273–283.

[3] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, John Wiley, New York, 1987.

[4] F. Harary, *The number of linear, directed, rooted, and connected graphs*, Trans. Amer. Math. Soc., 78 (1955), pp. 445–463.

[5] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, New York, 1973.

[6] M. Hofmeister, *Counting double covers of graphs*, J. Graph Theory, 12 (1988), pp. 437–444.

[7] M. Hofmeister, *Isomorphisms and automorphisms of coverings*, Discrete Math., 98 (1991), pp. 175–183.

[8] M. Hofmeister, *Concrete graph covering projections*, Ars Combin., 32 (1991), pp. 121–127.

[9] M. Hofmeister, *Enumeration of concrete regular covering projections*, SIAM J. Discrete Math., 8 (1995), pp. 51–61.

[10] M. Hofmeister, *Graph covering projections arising from finite vector spaces over finite fields*, Discrete Math., 143 (1995), pp. 87–97.

[11] S. Hong and J. H. Kwak, *Regular fourfold coverings with respect to the identity automorphism*, J. Graph Theory, 15 (1993), pp. 621–627.

[12] S. Hong, J. H. Kwak, and J. Lee, *Regular graph coverings whose covering transformation groups have the isomorphism extension property*, Discrete Math., 148 (1996), pp. 85–105.

[13] J. H. Kwak, J.-H. Chun, and J. Lee, *Enumeration of regular graph coverings having finite abelian covering transformation groups*, SIAM J. Discrete Math., 11 (1998), pp. 228–240.

[14] J. H. Kwak and J. Lee, *Isomorphism classes of graph bundles*, Canad. J. Math., XLII (1990), pp. 747–761.

[15] J. H. Kwak and J. Lee, *Counting some finite-fold coverings of a graph*, Graphs Combin., 8 (1992), pp. 277–285.

[16] J. H. Kwak and J. Lee, *Isomorphism classes of cycle permutation graphs*, Discrete Math., 105 (1992), pp. 131–142.

[17] J. H. Kwak and J. Lee, *Enumeration of graph coverings and its applications*, in Graph Theory, Combinatorics, Algorithms, and Applications: Proc. 7th Quadrennial International Conference on the Theory and Applications of Graphs, Y. Alavi and A. Schwenk, eds., John Wiley, New York, 1995, pp. 649–659.

[18] I. Sato, *Isomorphisms of some coverings*, Discrete Math., 128 (1994), pp. 317–326.

[19] I. Tomescu, *Problems in Combinatorics and Graph Theory*, John Wiley, New York, 1985.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 273–285, May 1998

007

ENUMERATION OF REGULAR GRAPH COVERINGS HAVING FINITE ABELIAN COVERING TRANSFORMATION GROUPS*

JIN HO KWAK†, JANG-HO CHUN‡, AND JAEUN LEE‡

Abstract. Several isomorphism classes of graph coverings of a graph G have been enumerated by many authors. An enumeration of the isomorphism classes of n-fold coverings of a graph G was done by Kwak and Lee [*Canad. J. Math.*, XLII (1990), pp. 747–761] and independently by Hofmeister [*Discrete Math.*, 98 (1991), pp. 437–444]. An enumeration of the isomorphism classes of connected n-fold coverings of a graph G was recently done by Kwak and Lee [*J. Graph Theory*, 23 (1996), pp. 105–109]. But the enumeration of the isomorphism classes of regular coverings of a graph G has been done for only a few cases. In fact, the isomorphism classes of $\mathcal{A}$-coverings of G were enumerated when $\mathcal{A}$ is the cyclic group $\mathbb{Z}_n$, the dihedral group $\mathbb{D}_n$ (n: odd), and the direct sum of m copies of $\mathbb{Z}_p$. (See [*Discrete Math.*, 143 (1995), pp. 87–97], [*J. Graph Theory*, 15 (1993), pp. 621–627], and [*Discrete Math.*, 148 (1996), pp. 85–105]).

In this paper, we discuss a method to enumerate the isomorphism classes of connected $\mathcal{A}$-coverings of a graph G for any finite group $\mathcal{A}$ and derive some formulas for enumerating the isomorphism classes of regular n-fold coverings for any natural number n. In particular, we calculate the number of the isomorphism classes of $\mathcal{A}$-coverings of G when $\mathcal{A}$ is a finite abelian group or the dihedral group $\mathbb{D}_n$. Our method gives partial answers to the open problems 1 and 2 in [*Discrete Math.*, 148 (1996), pp. 85–105] and also gives a formula to calculate the number of the subgroups of a given index of any finitely generated free abelian group.

Key words. regular coverings of a graph, (ordinary) voltage assignments, enumeration

AMS subject classifications. 05C10, 05C30, 20K27, 57M15

PII. S0895480196304428

1. Introduction.

Let G be a connected finite simple graph with vertex set $V(G)$ and edge set $E(G)$. The *neighborhood* of a vertex $v \in V(G)$, denoted by $N(v)$, is the set of vertices adjacent to v. We use $|X|$ for the cardinality of a set X. The number $\beta(G) = |E(G)| - |V(G)| + 1$ is equal to the number of independent cycles in G and it is referred to as the *Betti number* of G.

Two graphs G and H are *isomorphic* if there exists a one-to-one correspondence between their vertex sets which preserves adjacency, and such a correspondence is called an *isomorphism* between G and H. An *automorphism* of a graph G is an isomorphism of G onto itself. Thus, an automorphism of G is a permutation of the vertex set $V(G)$ which preserves adjacency. Obviously, a composition of two automorphisms is also an automorphism. Hence the automorphisms of G form a permutation group, $\mathrm{Aut}\,(G)$, which acts on the vertex set $V(G)$.

A graph $\widetilde{G}$ is called a *covering* of G with projection $p : \widetilde{G} \to G$ if there is a surjection $p : V(\widetilde{G}) \to V(G)$ such that $p|_{N(\tilde{v})} : N(\tilde{v}) \to N(v)$ is a bijection for any vertex $v \in V(G)$ and $\tilde{v} \in p^{-1}(v)$. We also say that the projection $p : \widetilde{G} \to G$ is an *n-fold covering* of G if p is n-to-one. A covering $p : \widetilde{G} \to G$ is said to be *regular* (simply, *$\mathcal{A}$-covering*) if there is a subgroup $\mathcal{A}$ of the automorphism group $\mathrm{Aut}\,(\widetilde{G})$ of $\widetilde{G}$ acting freely on $\widetilde{G}$ so that the graph G is isomorphic to the quotient graph $\widetilde{G}/\mathcal{A}$,

*Received by the editors May 28, 1996; accepted for publication (in revised form) March 10, 1997. http://www.siam.org/journals/sidma/11-2/30442.html

†Department of Mathematics, Pohang University of Science and Technology, Pohang 790-784, Korea (jinkwak@postech.ac.kr). The research of this author was supported by BSRI-96-1430.

‡Department of Mathematics, Yeungnam University, Kyongsan 712-749, Korea (jhchun@ynucc.yeungnam.ac.kr, julee@ynucc.yeungnam.ac.kr). The research of J. Lee was supported by TGRC-KOSEF and BSRI-1409.

say by h, and the quotient map $\widetilde{G} \to \widetilde{G}/\mathcal{A}$ is the composition $h \circ p$ of p and h. The fiber of an edge or a vertex is its preimage under p.

Two coverings $p_i : \widetilde{G}_i \to G$, $i = 1, 2$ are said to be *isomorphic* if there exists a graph isomorphism $\Phi : \widetilde{G}_1 \to \widetilde{G}_2$ such that the diagram

$$\begin{array}{ccc} \widetilde{G}_1 & \xrightarrow{\Phi} & \widetilde{G}_2 \\ & {\scriptstyle p_1}\searrow \quad \swarrow{\scriptstyle p_2} & \\ & G & \end{array}$$

commutes. Such a Φ is called a *covering isomorphism*.

Every edge of a graph G gives rise to a pair of oppositely directed edges. By $e^{-1} = vu$, we mean the reverse edge to a directed edge $e = uv$. We denote the set of directed edges of G by $D(G)$. Let $\mathcal{A}$ be a finite group. An *ordinary voltage assignment* (or, $\mathcal{A}$-*voltage assignment*) of G is a function $\phi : D(G) \to \mathcal{A}$ with the property that $\phi(e^{-1}) = \phi(e)^{-1}$ for each $e \in D(G)$. The values of ϕ are called *voltages*, and $\mathcal{A}$ is called the *voltage group*. The *ordinary derived graph* $G \times_\phi \mathcal{A}$ derived from an ordinary voltage assignment $\phi : D(G) \to \mathcal{A}$ has as its vertex set $V(G) \times \mathcal{A}$, and as its edge set $E(G) \times \mathcal{A}$, so that an edge (e, g) of $G \times_\phi \mathcal{A}$ joins a vertex (u, g) to $(v, \phi(e)g)$ for $e = uv \in D(G)$ and $g \in \mathcal{A}$. In the (ordinary) derived graph $G \times_\phi \mathcal{A}$, a vertex (u, g) is denoted by u_g and an edge (e, g) is denoted by e_g. The first coordinate projection $p_\phi : G \times_\phi \mathcal{A} \to G$, called the natural projection, commutes with the left multiplication action of the $\phi(e)$ and the right multiplication action of $\mathcal{A}$ on the fibers, which is free and transitive, so that p_ϕ is a regular $|\mathcal{A}|$-fold covering, called simply an $\mathcal{A}$-*covering*. Gross and Tucker [1] showed that every $\mathcal{A}$-covering $\widetilde{G}$ of a graph G can be derived from an $\mathcal{A}$-voltage assignment.

Kwak and Lee enumerated the isomorphism classes of n-fold coverings of a graph G as follows.

THEOREM 1.1 (see [9]). *The number* $\mathrm{Iso}\,(G; n)$ *of isomorphism classes of* n*-fold coverings of a connected graph* G *is*

$$\mathrm{Iso}\,(G; n) = \sum_{\ell_1 + 2\ell_2 + \cdots + n\ell_n = n} (\ell_1!\, 2^{\ell_2} \ell_2! \cdots n^{\ell_n} \ell_n!)^{\beta(G)-1}.$$

But the enumeration of the isomorphism classes of regular coverings of a graph has been done for only few cases. In fact, the isomorphism classes of $\mathcal{A}$-coverings of G were enumerated when $\mathcal{A}$ is the cyclic group $\mathbb{Z}_n$, the dihedral group $\mathbb{D}_n$ (n: odd), and the direct sum of m copies of $\mathbb{Z}_p$. (See [3]–[10], [14].)

In this paper, we derive formulas for enumerating the isomorphism classes of *regular* n-fold coverings for any natural number n and enumerate the isomorphism classes of $\mathcal{A}$-coverings of G when $\mathcal{A}$ is a finite abelian group or the dihedral group $\mathbb{D}_n$.

2. Enumeration of regular n-fold coverings. For a finite group $\mathcal{A}$, let $\mathrm{Iso}\,(G; \mathcal{A})$ (resp., $\mathrm{Isoc}\,(G; \mathcal{A})$) denote the number of the isomorphism classes of $\mathcal{A}$-coverings (resp., connected $\mathcal{A}$-coverings) of G. Let $\mathrm{Iso}^R(G; n)$ (resp., $\mathrm{Isoc}^R(G; n)$) denote the number of the isomorphism classes of *regular* n-fold coverings (resp., connected regular n-fold coverings) of G. The number $\mathrm{Isoc}\,(G; n)$ of the isomorphism classes of connected n-fold coverings of G was already calculated by Kwak and Lee (see [13]). In this section, we calculate the number $\mathrm{Iso}^R(G; n)$ of the isomorphism classes of regular n-fold coverings of G.

For a finite group $\mathcal{A}$, let $C^1(G;\mathcal{A})$ denote the set of all $\mathcal{A}$-voltage assignments ϕ of G. Let T be a spanning tree of G and let

$$C_T^1(G;\mathcal{A}) = \{\phi \in C^1(G;\mathcal{A}) : \phi(uv) = \text{identity} \quad \text{for each } uv \in D(T)\}.$$

It is known that every $\mathcal{A}$-covering of G can be derived from an $\mathcal{A}$-voltage assignment ϕ in $C_T^1(G;\mathcal{A})$ (see [1], [8]). From now on, let T denote a fixed spanning tree of a graph G, and we consider only an $\mathcal{A}$-voltage assignment ϕ in $C_T^1(G;\mathcal{A})$.

For a voltage assignment $\phi \in C_T^1(G;\mathcal{A})$, let $\mathcal{A}_\phi(v)$ denote the local voltage group of ϕ at v which is, by definition, the subgroup of $\mathcal{A}$ consisting of all net ϕ-voltages of the closed walks based at $v \in V(G)$. The net ϕ-voltage of a closed walk is the product of the forward voltages (written from right to left) along the edges of the walk. Clearly, the local voltage groups $\mathcal{A}_\phi(v)$ of $\phi \in C_T^1(G;\mathcal{A})$, $v \in V(G)$ are independent of the choice of the base vertex v, and we simply denote it by $\mathcal{A}_\phi$. It is clear by the definition of the ordinary derived graph $G \times_\phi \mathcal{A}$ that for any voltage assignment $\phi \in C_T^1(G;\mathcal{A})$, the derived graph $G \times_\phi \mathcal{A}$ is connected if and only if the local voltage group $\mathcal{A}_\phi$ is just the full group $\mathcal{A}$.

Hong, Kwak, and Lee obtained an algebraic characterization of two $\mathcal{A}$-coverings of a graph G to be isomorphic.

THEOREM 2.1 (see [8]). *Let ϕ and ψ be two voltage assignments in $C_T^1(G;\mathcal{A})$. Then two $\mathcal{A}$-coverings $p_\phi : G \times_\phi \mathcal{A} \to G$ and $p_\psi : G \times_\psi \mathcal{A} \to G$ are isomorphic if and only if there exists a group isomorphism $\sigma : \mathcal{A}_\phi \to \mathcal{A}_\psi$ such that*

$$\psi(uv) = \sigma(\phi(uv))$$

for all $uv \in D(G) - D(T)$ (or $\in D(G)$). Moreover, if both ϕ and ψ derive connected coverings, then it is also equivalent to say that there exists a group automorphism $\sigma \in \mathrm{Aut}(\mathcal{A})$ such that

$$\psi(uv) = \sigma(\phi(uv))$$

for all $uv \in D(G) - D(T)$ (or $\in D(G)$).

By using a method similar to the proof of Theorem 2.1, we can have the following theorem.

THEOREM 2.2. *Let $\mathcal{A}$ and $\mathcal{B}$ be two finite groups, and let $\phi \in C_T^1(G;\mathcal{A})$, $\psi \in C_T^1(G;\mathcal{B})$ be two voltage assignments. Then two coverings $p_\phi : G \times_\phi \mathcal{A} \to G$ and $p_\psi : G \times_\psi \mathcal{B} \to G$ are isomorphic if and only if there exists a group isomorphism $\sigma : \mathcal{A}_\phi \to \mathcal{B}_\psi$ such that*

$$\psi(uv) = \sigma(\phi(uv))$$

for all $uv \in D(G) - D(T)$ (or $\in D(G)$). Moreover, if both ϕ and ψ derive connected coverings, then it is also equivalent to say that there exists a group isomorphism $\sigma : \mathcal{A} \to \mathcal{B}$ such that

$$\psi(uv) = \sigma(\phi(uv))$$

for all $uv \in D(G) - D(T)$ (or $\in D(G)$).

Let $\phi \in C_T^1(G;\mathcal{A})$ be a voltage assignment. Then the covering $p_\phi : G \times_\phi \mathcal{A} \to G$ is regular, and each component of $G \times_\phi \mathcal{A}$ is isomorphic to the component of $G \times_\phi \mathcal{A}$ containing the vertices $\{\, v_{id} \,|\, v \in V(G) \,\}$, called the *identity component* of $G \times_\phi \mathcal{A}$, where id denotes the identity element of the group $\mathcal{A}$. In fact, the identity component

of an $\mathcal{A}$-covering $G \times_\phi \mathcal{A}$ is just the $\mathcal{A}_\phi$-covering $G \times_\phi \mathcal{A}_\phi$ by the construction of the derived graph. Now, it comes from Theorem 2.2 that two regular coverings of the same fold number of a graph are isomorphic if and only if their identity components are isomorphic as coverings. Notice that the order of any subgroup of a finite group $\mathcal{A}$ is a divisor of the order $|\mathcal{A}|$ of the group $\mathcal{A}$. Now we have the following theorem.

THEOREM 2.3. *For any natural number n,*

$$\mathrm{Iso}^R(G;n) = \sum_{d|n} \mathrm{Isoc}^R(G;d).$$

In particular, if $n = p^\ell$ for a prime p, then

$$\mathrm{Iso}^R(G;p^\ell) = \sum_{i=0}^{\ell} \mathrm{Isoc}^R(G;p^i).$$

From Theorem 2.2, we can see that $\mathrm{Iso}(G;\mathcal{A}) = \mathrm{Iso}(G;\mathcal{B})$ and $\mathrm{Isoc}(G;\mathcal{A}) = \mathrm{Isoc}(G;\mathcal{B})$ for any two isomorphic finite groups $\mathcal{A}$ and $\mathcal{B}$. Moreover, $\mathrm{Iso}(G;\mathcal{A})$ (resp., $\mathrm{Isoc}(G;\mathcal{A})$) is equal to the number of isomorphism classes of (resp., connected) regular $|\mathcal{A}|$-fold coverings of G whose covering transformation groups are isomorphic to the group $\mathcal{A}$. The following theorem comes from these facts and Theorem 2.2.

THEOREM 2.4. *For any natural number n,*

$$\mathrm{Isoc}^R(G;n) = \sum_{\mathcal{A}} \mathrm{Isoc}\,(G;\mathcal{A}),$$

where $\mathcal{A}$ runs over all representatives of isomorphism classes of groups of order n.

An analogous argument to the proof of Theorem 2.3 gives the following.

THEOREM 2.5. *For any finite group $\mathcal{A}$,*

$$\mathrm{Iso}\,(G;\mathcal{A}) = \sum_{\mathcal{S}} \mathrm{Isoc}\,(G;\mathcal{S}),$$

where $\mathcal{S}$ runs over all representatives of isomorphism classes of subgroups of $\mathcal{A}$.

By combining Theorems 2.3, 2.5, and 2.4, we can see that

$$\begin{aligned}
\mathrm{Iso}^R(G;p^2) &= \mathrm{Isoc}^R(G;p^2) + \mathrm{Isoc}^R(G;p) + \mathrm{Isoc}^R(G;1) \\
&= \mathrm{Isoc}\,(G;\mathbb{Z}_p \oplus \mathbb{Z}_p) + \mathrm{Isoc}\,(G;\mathbb{Z}_{p^2}) + \mathrm{Isoc}\,(G;\mathbb{Z}_p) + 1 \\
&= \mathrm{Iso}\,(G;\mathbb{Z}_p \oplus \mathbb{Z}_p) + \mathrm{Isoc}\,(G;\mathbb{Z}_{p^2}).
\end{aligned}$$

Now, we need to calculate the number $\mathrm{Isoc}\,(G;\mathcal{A})$ for any finite group $\mathcal{A}$. For a finite group $\mathcal{A}$ and a natural number n, let

$$\mathfrak{G}(\mathcal{A};n) = \{\, (g_1, g_2, \ldots, g_n) \in \mathcal{A}^n \,:\, \{g_1, g_2, \ldots, g_n\} \text{ generates } \mathcal{A}\}.$$

THEOREM 2.6. *For any finite group $\mathcal{A}$,*

$$\mathrm{Isoc}\,(G;\mathcal{A}) = \frac{|\mathfrak{G}(\mathcal{A};\beta(G))|}{|\mathrm{Aut}\,(\mathcal{A})|}.$$

Proof. It is clear that for any $\mathcal{A}$-voltage assignment ϕ in $C^1_T(G;\mathcal{A})$, the local voltage group $\mathcal{A}_\phi$ is the subgroup of $\mathcal{A}$ generated by voltages $\phi(e)$ on the edges e in the cotree $G - T$. Recall that the covering graph $G \times_\phi \mathcal{A}$ of G is connected if

and only if the local voltage group $\mathcal{A}_\phi$ is the full group $\mathcal{A}$. We also notice that the number of positively directed edges in $G - T$ is $|E(G - T)|$, which is equal to $\beta(G)$. By the definition of $\mathfrak{G}(\mathcal{A}; n)$, $\mathfrak{G}(\mathcal{A}; \beta(G))$ can be identified with the set of $\mathcal{A}$-voltage assignments of G whose derived coverings are all of the connected $\mathcal{A}$-coverings of G. Now, Theorem 2.6 comes from the Burnside lemma, because the $\mathrm{Aut}\,(\mathcal{A})$ action on $\mathfrak{G}(\mathcal{A}; n)$ given in Theorem 2.1 is free for each n. □

Example 1. Let $\mathbb{Z}_{p^m}$ be the cyclic group of order p^m, p prime. Then $\mathrm{Aut}\,(\mathbb{Z}_{p^m})$ can be identified with the set of all elements of $\mathbb{Z}_{p^m}$ which are relatively prime to p^m; that is, the set $\{\lambda \in \mathbb{Z}_{p^m} : (\lambda, p^m) = 1\}$, and

$$\mathfrak{G}(\mathbb{Z}_{p^m}; \beta(G)) = \{(g_1, g_2, \cdots, g_{\beta(G)}) \in (\mathbb{Z}_{p^m})^{\beta(G)} \mid \text{at least one of } g_i\text{'s generates } \mathbb{Z}_{p^m}\}.$$

It implies that

$$|\mathrm{Aut}\,(\mathbb{Z}_{p^m})| = p^{m-1}(p-1) \quad \text{and} \quad |\mathfrak{G}(\mathbb{Z}_{p^m}; \beta(G))| = p^{\beta(G)m} - p^{\beta(G)(m-1)}.$$

Then, by Theorem 2.6,

$$\mathrm{Isoc}\,(G; \mathbb{Z}_{p^m}) = \frac{p^{\beta(G)m} - p^{\beta(G)(m-1)}}{p^{m-1}(p-1)} = p^{(\beta(G)-1)(m-1)} \frac{p^{\beta(G)} - 1}{p-1}$$

for $m > 0$. By Theorem 2.5 and the lattice structure of subgroups of $\mathbb{Z}_{p^m}$, we have

$$\mathrm{Iso}\,(G; \mathbb{Z}_{p^m}) = 1 + \sum_{h=1}^{m} p^{(\beta(G)-1)(h-1)} \frac{p^{\beta(G)} - 1}{p-1} = 1 + \frac{p^{\beta(G)} - 1}{p-1} \, \frac{p^{m(\beta(G)-1)} - 1}{p^{\beta(G)-1} - 1}.$$

Note that the number $\mathrm{Iso}\,(G; \mathbb{Z}_p) = \mathrm{Iso}^R(G; p)$ was already calculated in [9].

Next, we derive a product formula for the isomorphism classes of (connected) regular coverings of a graph G.

THEOREM 2.7. *For any two finite groups $\mathcal{A}$ and $\mathcal{B}$ with $(|\mathcal{A}|, |\mathcal{B}|) = 1$,*

$$\mathrm{Isoc}\,(G; \mathcal{A} \oplus \mathcal{B}) = \mathrm{Isoc}\,(G; \mathcal{A})\ \mathrm{Isoc}\,(G; \mathcal{B}),$$

and

$$\mathrm{Iso}\,(G; \mathcal{A} \oplus \mathcal{B}) = \mathrm{Iso}\,(G; \mathcal{A})\ \mathrm{Iso}\,(G; \mathcal{B}).$$

Proof. For any $\phi \in C_T^1(G; \mathcal{A} \oplus \mathcal{B})$, we define $\phi_\mathcal{A} \in C_T^1(G; \mathcal{A})$ and $\phi_\mathcal{B} \in C_T^1(G; \mathcal{B})$ so that

$$\phi(e) = (\phi_\mathcal{A}(e),\ \phi_\mathcal{B}(e))$$

for any e in $D(G)$. If ϕ derives a connected covering, then the local voltage group $(\mathcal{A} \oplus \mathcal{B})_\phi$ is $\mathcal{A} \oplus \mathcal{B}$. Because $(|\mathcal{A}|, |\mathcal{B}|) = 1$, both $\phi_\mathcal{A}$ and $\phi_\mathcal{B}$ derive connected coverings. Conversely, if $\phi_1 \in C_T^1(G; \mathcal{A})$ and $\phi_2 \in C_T^1(G; \mathcal{B})$ drive connected coverings, then the voltage assignment $\phi \in C_T^1(G; \mathcal{A} \oplus \mathcal{B})$ defined by $\phi(e) = (\phi_1(e),\ \phi_2(e))$, $e \in D(G)$ derives a connected covering because $\mathcal{A}_{\phi_1} = \mathcal{A}$, $\mathcal{B}_{\phi_2} = \mathcal{B}$, and $(|\mathcal{A}|, |\mathcal{B}|) = 1$. It implies that

$$|\mathfrak{G}(\mathcal{A} \oplus \mathcal{B}; \beta(G))| = |\mathfrak{G}(\mathcal{A}; \beta(G))|\, |\mathfrak{G}(\mathcal{B}; \beta(G))|.$$

Since $\mathrm{Aut}\,(\mathcal{A} \oplus \mathcal{B}) = \mathrm{Aut}\,(\mathcal{A}) \oplus \mathrm{Aut}\,(\mathcal{B})$, it comes from Theorem 2.6 that

$$\mathrm{Isoc}\,(G; \mathcal{A} \oplus \mathcal{B}) = \mathrm{Isoc}\,(G; \mathcal{A})\ \mathrm{Isoc}\,(G; \mathcal{B}).$$

Notice that every subgroup of $\mathcal{A} \oplus \mathcal{B}$ is of the form $\mathcal{S}_1 \oplus \mathcal{S}_2$, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are subgroups of $\mathcal{A}$ and $\mathcal{B}$, respectively. Since $(|\mathcal{A}|, |\mathcal{B}|) = 1$, $(|\mathcal{S}_1|, |\mathcal{S}_2|) = 1$. Now, it comes from Theorem 2.5 that $\mathrm{Iso}\,(G; \mathcal{A} \oplus \mathcal{B}) = \mathrm{Iso}\,(G; \mathcal{A})\ \mathrm{Iso}\,(G; \mathcal{B})$. □

COROLLARY 2.8. *Let m and n be two relatively prime numbers. Then*

$$\mathrm{Iso}^R(G; mn) \geq \mathrm{Iso}^R(G; m)\,\mathrm{Iso}^R(G; n).$$

In particular, if p and q are two distinct prime numbers such that $p < q$ and $p \nmid (q-1)$, then

$$\mathrm{Iso}^R(G; pq) = \mathrm{Iso}^R(G; p)\,\mathrm{Iso}^R(G; q).$$

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be two groups of order m and n, respectively. Then $\mathcal{A} \oplus \mathcal{B}$ is a group of order mn. This implies that

$$\mathrm{Iso}^R(G; mn) \geq \mathrm{Iso}^R(G; m)\,\mathrm{Iso}^R(G; n).$$

Note that if p and q are two distinct prime numbers such that $p < q$ and $p \nmid (q-1)$, then every group of order pq is isomorphic to the cyclic group $\mathbb{Z}_{pq}$ of order pq (see [16]). Hence,

$$\mathrm{Iso}^R(G; pq) = \mathrm{Iso}\,(G; \mathbb{Z}_{pq}) = \mathrm{Iso}\,(G; \mathbb{Z}_p)\,\mathrm{Iso}\,(G; \mathbb{Z}_q) = \mathrm{Iso}^R(G; p)\,\mathrm{Iso}^R(G; q).$$

This completes the proof. □

Remark. The number $\mathrm{Iso}^R(G; mn)$ can be strictly greater than the number $\mathrm{Iso}^R(G; m)\,\mathrm{Iso}^R(G; n)$, even if m and n are distinct primes. For example, if $\beta(G) \geq 2$, $m = 2$, and $n = 3$, then $\mathrm{Iso}^R(G; 6) > \mathrm{Iso}^R(G; 2)\,\mathrm{Iso}^R(G; 3)$, because

$$\begin{aligned} \mathrm{Iso}^R(G; 6) &= \mathrm{Isoc}\,(G; \mathbb{Z}_6) + \mathrm{Isoc}\,(G; \mathbb{D}_3) + \mathrm{Isoc}\,(G; \mathbb{Z}_2) + \mathrm{Isoc}\,(G; \mathbb{Z}_3) + 1 \\ &= \mathrm{Iso}\,(G; \mathbb{Z}_6) + \mathrm{Isoc}\,(G; \mathbb{D}_3), \end{aligned}$$

and

$$\mathrm{Iso}\,(G; \mathbb{Z}_6) = \mathrm{Iso}\,(G; \mathbb{Z}_2)\,\mathrm{Iso}\,(G; \mathbb{Z}_3) = \mathrm{Iso}^R(G; 2)\,\mathrm{Iso}^R(G; 3).$$

3. Enumeration of $\mathcal{A}$-coverings; $\mathcal{A}$ = abelian group. In [5], Hofmeister gave a formula for calculating the number $\mathrm{Iso}\,(G; m\mathbb{Z}_p)$, where $m\mathbb{Z}_p$ is the m-dimensional vector space over the finite field $\mathbb{Z}_p$. In this section, we calculate the number $\mathrm{Iso}\,(G; \mathcal{A})$ for any finite abelian group $\mathcal{A}$, which is much simpler and more explicit than Hofmeister's, when $\mathcal{A} = m\mathbb{Z}_p$.

By the classification of finite abelian groups, any finite abelian group $\mathcal{A}$ is isomorphic to a direct sum of finite cyclic groups of order powers of prime numbers. In order to calculate the number $\mathrm{Iso}\,(G; \mathcal{A})$, it suffices, by Theorems 2.5 and 2.7, to calculate the number $\mathrm{Iso}\,(G; \oplus_{h=1}^{\ell} m_h \mathbb{Z}_{p^{s_h}})$ or the number $\mathrm{Isoc}\,(G; \oplus_{h=1}^{\ell} m_h \mathbb{Z}_{p^{s_h}})$ for a prime p. To do this, we start with the following lemma.

LEMMA 3.1.

1. *For any natural numbers m and n with $m \leq n$, and a prime p, we have*

$$|\mathfrak{G}(m\mathbb{Z}_p; n)| = p^{\frac{m(m-1)}{2}}(p^n - 1)(p^{n-1} - 1)\cdots(p^{n-m+1} - 1),$$

and

$$|\mathrm{Aut}\,(m\mathbb{Z}_p)| = |\mathfrak{G}(m\mathbb{Z}_p; m)| = p^{\frac{m(m-1)}{2}}(p^m - 1)(p^{m-1} - 1)\cdots(p - 1).$$

2. *For any natural number $s \geq 1$, we have*

$$|\mathfrak{G}(m\mathbb{Z}_{p^s}; n)| = p^{(s-1)mn}|\mathfrak{G}(m\mathbb{Z}_p; n)|,$$

and

$$|\mathrm{Aut}\,(m\mathbb{Z}_{p^s})| = p^{(s-1)m^2}|\mathrm{Aut}\,(m\mathbb{Z}_p)|.$$

Proof. 1. To calculate the number $|\mathfrak{G}(m\mathbb{Z}_p; n)|$, it suffices to find a one-to-one correspondence between the set $\mathfrak{G}(m\mathbb{Z}_p; n)$ and the set of all linearly independent m ordered vectors in $n\mathbb{Z}_p$, because the cardinality of the set of all linearly independent m ordered vectors in $n\mathbb{Z}_p$ is $(p^n-1)(p^n-p), \ldots, (p^n-p^{m-1})$. For each $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ in $\mathfrak{G}(m\mathbb{Z}_p; n)$, let A be the $m \times n$ matrix having $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ as column vectors. Then each row of the matrix A is a vector in $n\mathbb{Z}_p$ and the rows of A form a linearly independent m ordered vectors in $n\mathbb{Z}_p$, because the rank of A is m. This completes the calculation of $|\mathfrak{G}(m\mathbb{Z}_p; n)|$. Since there is a one-to-one correspondence between $\mathrm{Aut}\,(m\mathbb{Z}_p)$ and the group of all linear automorphisms on the vector space $m\mathbb{Z}_p$ over the field $\mathbb{Z}_p$, we get $|\mathrm{Aut}\,(m\mathbb{Z}_p)| = |\mathfrak{G}(m\mathbb{Z}_p; m)|$.

2. Let $\mathbb{Z}_{p^s} =< 1 >$, where 1 is a generator of the additive cyclic group $\mathbb{Z}_{p^s}$. Then the element p^{s-1} in $\mathbb{Z}_{p^s}$ generates the cyclic subgroup $\mathbb{Z}_p$ of the group $\mathbb{Z}_{p^s}$, and an element γ in $\mathbb{Z}_{p^s}$ generates the group $\mathbb{Z}_{p^s}$ if and only if $p^{s-1}\gamma$ generates the subgroup $\mathbb{Z}_p$. It implies that any n elements $g_1, \ldots, g_n$ in the group $m\mathbb{Z}_{p^s}$ generate the group $m\mathbb{Z}_{p^s}$ if and only if the n elements $p^{s-1}g_1, \ldots, p^{s-1}g_n$ generate the subgroup $m\mathbb{Z}_p$; that is,

$$\begin{aligned}\mathfrak{G}(m\mathbb{Z}_{p^s}; n) &= \{(g_1, \ldots, g_n) \in (m\mathbb{Z}_{p^s})^n \mid \{g_1, \ldots, g_n\} \text{ generates } m\mathbb{Z}_{p^s}\} \\ &= \{(g_1, \ldots, g_n) \in (m\mathbb{Z}_{p^s})^n \mid \{p^{s-1}g_1, \ldots, p^{s-1}g_n\} \text{ generates } m\mathbb{Z}_p\}.\end{aligned}$$

Since the map $\theta : m\mathbb{Z}_{p^s} \to m\mathbb{Z}_p$ defined by $\theta(\gamma_1, \ldots, \gamma_m) = (p^{s-1}\gamma_1, \ldots, p^{s-1}\gamma_m)$ is a group homomorphism and its kernel, $\mathbf{Ker}(\theta)$, is generated by the m elements $(p, 0, \ldots, 0), (0, p, \ldots, 0), \ldots, (0, 0, \ldots, p)$, we get $|\mathbf{Ker}(\theta)| = p^{(s-1)m}$ and

$$|\mathfrak{G}(m\mathbb{Z}_{p^s}; n)| = |\mathbf{Ker}(\theta)|^n\,|\mathfrak{G}(m\mathbb{Z}_p; n)| = p^{(s-1)mn}|\mathfrak{G}(m\mathbb{Z}_p; n)|.$$

Finally, we note that $m\mathbb{Z}_{p^s}$ is an m-dimensional free module over the ring $\mathbb{Z}_{p^s}$. To count the number $|\mathrm{Aut}\,(m\mathbb{Z}_{p^s})|$, let $\mathbf{e}_1, \ldots, \mathbf{e}_m$ denote the standard basis for the free module $m\mathbb{Z}_{p^s}$. Then the map $\theta : \mathrm{Aut}\,(m\mathbb{Z}_{p^s}) \to \mathfrak{G}(m\mathbb{Z}_{p^s}; m)$ defined by $\theta(\sigma) = (\sigma(\mathbf{e}_1), \ldots, \sigma(\mathbf{e}_m))$ is clearly bijective. Hence,

$$|\mathrm{Aut}\,(m\mathbb{Z}_{p^s})| = |\mathfrak{G}(m\mathbb{Z}_{p^s}; m)| = p^{(s-1)m^2}|\mathfrak{G}(m\mathbb{Z}_p; m)| = p^{(s-1)m^2}|\mathrm{Aut}\,(m\mathbb{Z}_p)|$$

by the second equation of 1. □

By Theorems 2.5, 2.6, and Lemma 3.1, we have the following corollary.

COROLLARY 3.2. *For any m, the number of the isomorphism classes of connected $m\mathbb{Z}_p$-coverings of G is*

$$\mathrm{Isoc}\,(G; m\mathbb{Z}_p) = \frac{(p^{\beta(G)}-1)(p^{\beta(G)-1}-1)\cdots(p^{\beta(G)-m+1}-1)}{(p^m-1)(p^{m-1}-1)\cdots(p-1)},$$

and the number of the isomorphism classes of $m\mathbb{Z}_p$-coverings of G is

$$\begin{aligned}\mathrm{Iso}\,(G; m\mathbb{Z}_p) &= 1 + \sum_{h=1}^{m} \frac{(p^{\beta(G)}-1)(p^{\beta(G)-1}-1)\cdots(p^{\beta(G)-h+1}-1)}{(p^h-1)(p^{h-1}-1)\cdots(p-1)} \\ &= 1 + \frac{p^{\beta(G)}-1}{p-1}\left(1 + \frac{p^{\beta(G)-1}-1}{p^2-1}\left(1 + \cdots\left(1 + \frac{p^{\beta(G)-m+1}-1}{p^m-1}\right)\cdots\right)\right).\end{aligned}$$

Notice that the calculating formula for the number $\mathrm{Iso}\,(G; m\mathbb{Z}_p)$ in Corollary 3.2 is much more explicit than that of Hofmeister's in [5].

Remark. It is well known (see [17]) that the number of the m-dimensional subspaces of the n-dimensional vector space $n\mathbb{Z}_p$ over the field $\mathbb{Z}_p$ is equal to the Gaussian coefficient

$$\left[\begin{array}{c} n \\ m \end{array} \right]_p = \frac{\prod_{i=n-m+1}^{n} (p^i - 1)}{\prod_{i=1}^{m} (p^i - 1)}.$$

Hence, we can say that the number of the isomorphism classes of connected $m\mathbb{Z}_p$-coverings of a graph G is equal to the number of the m-dimensional subspaces of the $\beta(G)$-dimensional vector space $\beta(G)\mathbb{Z}_p$.

Let $m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}}$ be the direct sum of two abelian groups $m_1\mathbb{Z}_{p^{s_1}}$ and $m_2\mathbb{Z}_{p^{s_2}}$ (say, $s_2 < s_1$) and let $g_1 = (g_{11}, g_{12}), \ldots, g_n = (g_{n1}, g_{n2}) \in m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}}$. Then $\{g_1, \ldots, g_n\}$ generates $m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}}$ if and only if $\{(p^{s_1-1}g_{11}, p^{s_2-1}g_{12}), \ldots, (p^{s_1-1}g_{n1}, p^{s_2-1}g_{n2})\}$ generates $(m_1 + m_2)\mathbb{Z}_p$. An analogous argument to the proof of Lemma 3.1 gives

$$|\mathfrak{G}(m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}}; n)| = p^{n\left(\, m_1(s_1-1)+m_2(s_2-1)\right)} |\mathfrak{G}((m_1 + m_2)\mathbb{Z}_p; n)|.$$

But, in general,

$$|\mathrm{Aut}\,(m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}})| \neq |\mathfrak{G}(m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}}; m_1 + m_2)|.$$

In fact, a group-theoretic exercise gives

$$|\mathrm{Aut}\,(m_1\mathbb{Z}_{p^{s_1}} \oplus m_2\mathbb{Z}_{p^{s_2}})| = p^{g(m_i,s_i)} \prod_{i=1}^{2} \prod_{h=1}^{m_i} \left(p^{m_i-h+1} - 1\right),$$

where

$$g(m_i, s_i) = m \left(\sum_{i=1}^{2} m_i(s_i - 1) \right) - m_1m_2(s_1 - s_2 - 1) + \frac{m(m-1)}{2}$$

with $m = m_1 + m_2$ and $s_2 < s_1$. In general, we have the following.

LEMMA 3.3. *Let $m_1, \ldots, m_\ell$ and $s_1, \ldots, s_\ell$ be natural numbers with $s_\ell < \cdots < s_1$. Let p be a prime number. Then we have*

1. $\left|\mathfrak{G}\left(\oplus_{h=1}^{\ell} m_h\mathbb{Z}_{p^{s_h}}; n\right)\right| = p^{n\left(\, m_1(s_1-1)+\cdots+m_\ell(s_\ell-1)\right)} |\mathfrak{G}((m_1 + \cdots + m_\ell)\mathbb{Z}_p; n)|.$
2. $\left|\mathrm{Aut}\,(\oplus_{h=1}^{\ell} m_h\mathbb{Z}_{p^{s_h}})\right| = p^{g(m_i,s_i)} \prod_{i=1}^{\ell} \prod_{h=1}^{m_i} \left(p^{m_i-h+1} - 1\right)$, *where*

$$g(m_i, s_i) = m \left(\sum_{i=1}^{\ell} m_i(s_i - 1) \right) - \sum_{i=1}^{\ell-1} m_i \left(\sum_{j=i+1}^{\ell} m_j(s_i - s_j - 1) \right) + \frac{m(m-1)}{2}$$

with $m = m_1 + \cdots + m_\ell$.

TABLE 3.1
The numbers Isoc *and* Iso *for small* (p,q) *and small* $\beta(G)$.

		Isoc			Iso		
β	(p,q)	$\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p$	$\mathbb{Z}_{q^2}$	$\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p\oplus\mathbb{Z}_{q^2}$	$\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p$	$\mathbb{Z}_{q^2}$	$\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p\oplus\mathbb{Z}_{q^2}$
1	$(2,3)$	0	1	0	4	3	12
2	$(2,5)$	6	30	180	32	37	1184
3	$(3,5)$	1404	775	1088100	2757	807	2224899
4	$(3,7)$	126360	137200	1695792000	161451	137601	22215819051
5	$(5,7)$	9518437500	6725201	64013405393437500	9533687306	6728003	64142676795829918

Now, the following comes from Theorem 2.6 and Lemma 3.3.

THEOREM 3.4. *Let* $m_1,\dots,m_\ell$ *and* $s_1,\dots,s_\ell$ *be natural numbers with* $s_\ell<\cdots<s_1$. *Then the number of the isomorphism classes of connected* $\oplus_{h=1}^{\ell} m_h\mathbb{Z}_{p^{s_h}}$*-coverings of* G *is*

$$\mathrm{Isoc}\left(G;\oplus_{h=1}^{\ell} m_h\mathbb{Z}_{p^{s_h}}\right)=p^{f(\beta(G),m_i,s_i)}\frac{\displaystyle\prod_{i=1}^{m}p^{\beta(G)-i+1}-1}{\displaystyle\prod_{j=1}^{\ell}\prod_{h=1}^{m_j}p^{m_j-h+1}-1},$$

where $m=m_1+\cdots+m_\ell$, p *is prime and*

$$f(\beta(G),m_i,s_i)=(\beta(G)-m)\left(\sum_{i=1}^{\ell}m_i(s_i-1)\right)+\sum_{i=1}^{\ell-1}m_i\left(\sum_{j=i+1}^{\ell}m_j(s_i-s_j-1)\right).$$

Now, we can calculate the number $\mathrm{Iso}\,(G;\mathcal{A})$ for any finite abelian group $\mathcal{A}$ by using Theorems 2.5, 2.7, and 3.4 repeatedly if necessary. For example, if p and q are two distinct prime numbers, then

$$\begin{aligned}
&\mathrm{Iso}\left(G;\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p\oplus\mathbb{Z}_{q^2}\right)\\
&\quad=\mathrm{Iso}\left(G;\mathbb{Z}_{p^3}\oplus\mathbb{Z}_p\right)\mathrm{Iso}\left(G;\mathbb{Z}_{q^2}\right)\\
&\quad=\left(1+\sum_{i=1}^{3}\mathrm{Isoc}\left(G;\mathbb{Z}_{p^i}\right)+\sum_{i=1}^{3}\mathrm{Isoc}\left(G;\mathbb{Z}_{p^i}\oplus\mathbb{Z}_p\right)\right)\left(1+\sum_{i=1}^{2}\mathrm{Isoc}\left(G;\mathbb{Z}_{q^i}\right)\right)\\
&\quad=\left(1+\frac{p^{\beta(G)}-1}{p-1}\left(1+p^{\beta(G)-1}\left(1+p^{\beta(G)-1}\right)\right)\right.\\
&\qquad\left.+\frac{\left(p^{\beta(G)}-1\right)\left(p^{\beta(G)-1}-1\right)}{(p-1)\left(p^2-1\right)}\left(1+p^{\beta(G)-2}(p+1)\left(1+p^{\beta(G)-1}\right)\right)\right)\\
&\qquad\times\left(1+\frac{\left(q^{\beta(G)}-1\right)\left(q^{\beta(G)-1}+1\right)}{q-1}\right).
\end{aligned}$$

For some abelian groups $\mathcal{A}$ and small $\beta(G)$, the numbers $\mathrm{Isoc}\,(G;\mathcal{A})$ and $\mathrm{Iso}\,(G;\mathcal{A})$ are listed in Table 3.1.

Remark. For a connected $\mathcal{A}$-covering $p:\tilde{G}\to G$, the image $p_*(\pi_1(\tilde{G}))$ of the fundamental group of the covering graph $\tilde{G}$ is a normal subgroup of the fundamental group $\pi_1(G)$ of the base graph G, and the quotient group $\pi_1(G)/p_*(\pi_1(\tilde{G}))$ is

isomorphic to $\mathcal{A}$. If $\mathcal{A}$ is abelian, then $p_*(\pi_1(\tilde{G}))$ contains the commutator subgroup $[\pi_1(G), \pi_1(G)]$ of the free group $\pi_1(G)$ generated by $\beta(G)$ elements. Since $[\pi_1(G), \pi_1(G)]$ is a normal subgroup of $\pi_1(G)$, the natural homomorphism $q : \pi_1(G) \to \pi_1(G)/[\pi_1(G), \pi_1(G)]$ induces a one-to-one correspondence between the set of all subgroups of $\pi_1(G)$ containing $[\pi_1(G), \pi_1(G)]$ and the set of all subgroups of the quotient group $\pi_1(G)/[\pi_1(G), \pi_1(G)]$. Notice that $\pi_1(G)/[\pi_1(G), \pi_1(G)]$ is the free abelian group generated by $\beta(G)$ elements. Now, from a well-known classification theorem for regular coverings of a topological space, it follows that the number $\sum_{\mathcal{A}} \mathrm{Isoc}^R(G; \mathcal{A})$, where $\mathcal{A}$ runs over all representatives of isomorphism classes of abelian groups of order n, is equal to the number of subgroups of index n of the free abelian group generated by $\beta(G)$ elements.

4. Enumeration of $\mathbb{D}_n$-coverings. Recall that the dihedral group of order $2n$ can be presented as follows:

$$\mathbb{D}_n = \left\langle a, b \,:\, a^2 = 1 = b^n, aba = b^{-1} \right\rangle.$$

In [8], Hong, Kwak, and Lee calculated the number $\mathrm{Iso}\,(G; \mathbb{D}_n)$ for *odd* $n \geq 3$. As an extension of their results, we now calculate the number $\mathrm{Iso}\,(G; \mathbb{D}_n)$ for any $n \geq 3$.

Note that $\mathbb{D}_1 = \mathbb{Z}_2$, $\mathbb{D}_2 = \mathbb{Z}_2 \oplus \mathbb{Z}_2$, $\mathbb{D}_n$ is not abelian for $n \geq 3$ with $\langle a \rangle = \mathbb{Z}_2$ and $\langle b \rangle = \mathbb{Z}_n$, and an element of $\mathbb{D}_n$ can be of the form b^i or ab^i for some $0 \leq i \leq n-1$.

LEMMA 4.1. *Let n be a natural number with prime decomposition $p_1^{m_1} \cdots p_\ell^{m_\ell}$. Then*

1. $|\mathrm{Aut}(\mathbb{D}_n)| = n\, p_1^{m_1-1}(p_1 - 1) \cdots p_\ell^{m_\ell - 1}(p_\ell - 1)$.
2. *For any natural number r,*

$$|\mathfrak{G}(\mathbb{D}_n; r)| = (2^r - 1) \prod_{i=1}^{\ell} p_i^{(m_i-1)r+1} \left(p_i^{r-1} - 1\right).$$

Proof. It is not hard to show that

$$\mathrm{Aut}(\mathbb{D}_n) = \left\{ \sigma_j^i \,:\, \sigma_j^i(a) = ab^i, \sigma_j^i(b) = b^j,\ 0 \leq i, j \leq n-1,\ (n, j) = 1 \right\}.$$

It implies that $|\mathrm{Aut}(\mathbb{D}_n)| = n\, p_1^{m_1-1}(p_1 - 1) \cdots p_\ell^{m_\ell-1}(p_\ell - 1)$.

Next, we calculate the number $|\mathfrak{G}(\mathbb{D}_n; r)|$. Since the prime decomposition of n is $p_1^{m_1} \cdots p_\ell^{m_\ell}$, $\mathbb{Z}_n =< b >$ is isomorphic to $\oplus_{i=1}^{\ell} \mathbb{Z}_{p_i^{m_i}}$, where $\mathbb{Z}_{p_i^{m_i}} =< b_i >$ with $b = b_1 \cdots b_\ell$. Note that $\mathbb{D}_n = \mathbb{Z}_n \cup a\mathbb{Z}_n$, disjoint union. It is clear that if $(g_1, \ldots, g_r) \in \mathfrak{G}(\mathbb{D}_n; r)$, then there exists at least one j $(1 \leq j \leq r)$ such that $g_j \in a\mathbb{Z}_n = \{ab^i \,|\, i = 1, \ldots, n\}$. Given any nonempty subset S of $\{1, 2, \ldots, r\}$, let $\mathfrak{G}[S]$ denote the set

$$\{(g_1, \ldots, g_r) \in \mathfrak{G}(\mathbb{D}_n; r) \,:\, g_j \in a\mathbb{Z}_n \text{ for } j \in S \text{ and } g_j \in \mathbb{Z}_n \text{ for } j \notin S\}.$$

Then

$$\bigcup_{S(\neq\emptyset) \subset \{1,2,\ldots,r\}} \mathfrak{G}[S] = \mathfrak{G}(\mathbb{D}_n; r).$$

Moreover, $\mathfrak{G}[S]$ and $\mathfrak{G}[T]$ are disjoint for any two distinct nonempty subsets S and T of $\{1, 2, \ldots, r\}$. It implies that

$$|\mathfrak{G}(\mathbb{D}_n; r)| = \left| \bigcup_{S(\neq\emptyset) \subset \{1,2,\ldots,r\}} \mathfrak{G}[S] \right| = \sum_{S(\neq\emptyset) \subset \{1,2,\ldots,r\}} |\mathfrak{G}[S]|.$$

For convenience, for each $g \in \mathbb{D}_n$, let

$$g = \begin{cases} (g'_1, \ldots, g'_\ell) & \text{if } g \in \mathbb{Z}_n = \oplus_{i=1}^{\ell} \mathbb{Z}_{p_i^{m_i}} \\ a(g'_1, \ldots, g'_\ell) & \text{if } g \in a\mathbb{Z}_n = a \oplus_{i=1}^{\ell} \mathbb{Z}_{p_i^{m_i}}. \end{cases}$$

Let S be a nonempty subset of $\{1, \ldots, r\}$ and $(g_1, \ldots, g_r) \in (\mathbb{D}_n)^r \equiv \prod_{i=1}^{r} \mathbb{D}_n$. Then $(g_1, \ldots, g_r) \in \mathfrak{G}[S]$ if and only if for each $i = 1, \cdots, \ell$,

$$(g'_{1_i}, \ldots, g'_{r_i}) \in \prod_{i=1}^{r} \mathbb{Z}_{p_i^{m_i}} - \bigcup_{k=0}^{p_i-1} \left(\prod_{j \notin S} \mathbb{Z}_{p_i^{m_i-1}} \times \prod_{j \in S} b_i^k \mathbb{Z}_{p_i^{m_i-1}} \right),$$

where $\mathbb{Z}_{p_i^{m_i-1}}$ is the subgroup of $\mathbb{Z}_{p_i^{m_i}}$ generated by $b_i^{p_i}$. It implies that for any nonempty subset S of $\{1, 2, \ldots, r\}$,

$$|\mathfrak{G}[S]| = \prod_{i=1}^{\ell} \left(p_i^{m_i r} - p_i^{(m_i-1)|S|} \cdot p_i \cdot p_i^{(m_i-1)(r-|S|)} \right) = \prod_{i=1}^{\ell} p_i^{(m_i-1)r+1} \left(p_i^{r-1} - 1 \right),$$

which does not depend on the set S. Now, the cardinality $|\mathfrak{G}(\mathbb{D}_n; r)|$ of the set $\mathfrak{G}(\mathbb{D}_n; r)$ is

$$\sum_{S(\neq\emptyset) \subset \{1,2,\cdots,r\}} |\mathfrak{G}[S]| = (2^r - 1) \prod_{i=1}^{\ell} p_i^{(m_i-1)r+1} \left(p_i^{r-1} - 1 \right). \qquad \square$$

Notice that any subgroup of the dihedral group $\mathbb{D}_n$ is isomorphic to one of $\mathbb{D}_i$ (i is a divisor of n) and $\mathbb{Z}_j$ (j is a divisor of n), where $\mathbb{Z}_1 = \{identity\}$. It follows from Theorem 2.5 that for any $n \geq 3$,

$$\begin{aligned} \mathrm{Iso}\,(G; \mathbb{D}_n) &= \begin{cases} \sum_{m|n} \mathrm{Isoc}\,(G; \mathbb{Z}_m) + \sum_{m|n} \mathrm{Isoc}\,(G; \mathbb{D}_m) & \text{if } n \text{ is odd,} \\ \sum_{m|n} \mathrm{Isoc}\,(G; \mathbb{Z}_m) + \sum_{m|n,\, m\neq 1} \mathrm{Isoc}\,(G; \mathbb{D}_m) & \text{if } n \text{ is even,} \end{cases} \\ &= \begin{cases} \mathrm{Iso}\,(G; \mathbb{Z}_n) + \sum_{m|n} \mathrm{Isoc}\,(G; \mathbb{D}_m) & \text{if } n \text{ is odd,} \\ \mathrm{Iso}\,(G; \mathbb{Z}_n) + \sum_{m|n,\, m\neq 1} \mathrm{Isoc}\,(G; \mathbb{D}_m) & \text{if } n \text{ is even.} \end{cases} \end{aligned}$$

First, we calculate the number $\mathrm{Isoc}\,(G; \mathbb{D}_n)$ for any $n \geq 3$. The following comes from Theorem 2.6 and Lemma 4.1.

THEOREM 4.2. *For any $n \geq 3$, the number of the isomorphism classes of connected $\mathbb{D}_n$-coverings of G is*

$$\mathrm{Isoc}\,(G; \mathbb{D}_n) = \left(2^{\beta(G)} - 1 \right) \prod_{i=1}^{\ell} p_i^{(m_i-1)(\beta(G)-2)} \frac{p_i^{\beta(G)-1} - 1}{p_i - 1},$$

where $p_1^{m_1} \cdots p_\ell^{m_\ell}$ is the prime decomposition of n.

For any edge e in the cotree $G-T$, we have $\beta(G-e)=\beta(G)-1$. By Example 1 and Theorems 3.4 and 4.2, we have

$$\mathrm{Isoc}\,(G;\mathbb{D}_n)=(2^{\beta(G)}-1)\mathrm{Isoc}\,(G-e;\mathbb{Z}_n)$$

for any $n\geq 3$. Thus, if n is odd, then

$$\sum_{m|n}\mathrm{Isoc}\,(G;\mathbb{D}_m)=\left(2^{\beta(G)}-1\right)\sum_{m|n}\mathrm{Isoc}\,(G-e;\mathbb{Z}_m)=\left(2^{\beta(G)}-1\right)\mathrm{Iso}\,(G-e;\mathbb{Z}_n).$$

If n is even, then

$$\begin{aligned}
&\sum_{m|n,\,m\neq 1}\mathrm{Isoc}\,(G;\mathbb{D}_m)\\
&\quad=\sum_{m|n,\,m\geq 3}\mathrm{Isoc}\,(G;\mathbb{D}_m)+\mathrm{Isoc}\,(G;\mathbb{D}_2)\\
&\quad=\left(2^{\beta(G)}-1\right)\left(\sum_{m|n}\mathrm{Isoc}\,(G-e;\mathbb{Z}_m)-[1+\mathrm{Isoc}\,(G-e;\mathbb{Z}_2)]\right)+\mathrm{Isoc}\,(G;\mathbb{D}_2)\\
&\quad=\left(2^{\beta(G)}-1\right)\mathrm{Iso}\,(G-e;\mathbb{Z}_n)-\left(2^{\beta(G)}-1\right)2^{\beta(G)-1}\\
&\qquad\qquad+\frac{1}{3}\left(2^{\beta(G)}-1\right)\left(2^{\beta(G)-1}-1\right)\\
&\quad=\left(2^{\beta(G)}-1\right)\mathrm{Iso}\,(G-e;\mathbb{Z}_n)-\frac{1}{3}\left(4^{\beta(G)}-1\right).
\end{aligned}$$

We summarize our discussion as follows.

THEOREM 4.3. *For any $n\geq 3$, the number of the isomorphism classes of $\mathbb{D}_n$-covering of G is*

$$\mathrm{Iso}\,(G;\mathbb{D}_n)=\begin{cases}\mathrm{Iso}\,(G;\mathbb{Z}_n)+\left(2^{\beta(G)}-1\right)\mathrm{Iso}\,(G-e;\mathbb{Z}_n) & \text{if } n \text{ is odd},\\ \mathrm{Iso}\,(G;\mathbb{Z}_n)+\left(2^{\beta(G)}-1\right)\mathrm{Iso}\,(G-e;\mathbb{Z}_n)\\ \qquad\qquad-\dfrac{1}{3}\left(4^{\beta(G)}-1\right) & \text{if } n \text{ is even},\end{cases}$$

where e is an edge in the cotree $G-T$, and

$$\mathrm{Iso}\,(G;\mathbb{Z}_n)=\begin{cases}1 & \text{if } \beta(G)=0,\\ \displaystyle\prod_{i=1}^{\ell}(m_i+1) & \text{if } \beta(G)=1,\\ \displaystyle\prod_{i=1}^{\ell}\left(1+\frac{p_i^{\beta(G)}-1}{p_i-1}\frac{p_i^{m_i(\beta(G)-1)}-1}{p_i^{\beta(G)-1}-1}\right) & \text{if } \beta(G)\geq 2,\end{cases}$$

where the prime decomposition of n is $p_1^{m_1}p_2^{m_2}\cdots p_\ell^{m_\ell}$.

The number $\mathrm{Iso}\,(G;\mathbb{Z}_n)$ comes from Theorem 2.7 and Example 1. We note that the numbers $\mathrm{Iso}\,(G;\mathbb{D}_n)$ ($n=$ odd) and $\mathrm{Iso}\,(G;\mathbb{Z}_n)$ were already counted in [8], but the calculating method in [8] is different from that in this paper. The numbers $\mathrm{Isoc}\,(G;\mathbb{D}_n)$ and $\mathrm{Iso}\,(G;\mathbb{D}_n)$ for small n and $\beta(G)$ are listed in Tables 4.1 and 4.2.

TABLE 4.1
The number $\mathrm{Isoc}\,(G;\mathbb{D}_n)$.

β	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$
1	0	0	0	0	0	0	0	0	0
2	3	3	3	3	3	3	3	3	3
3	28	42	42	84	56	84	84	126	84
4	195	420	465	1365	855	1680	1755	3255	1995
5	1240	3720	4836	18600	12400	29760	33480	72540	45384

TABLE 4.2
The number $\mathrm{Iso}\,(G;\mathbb{D}_n)$.

β	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$
1	3	3	3	4	3	4	4	4	3
2	11	14	13	27	15	29	26	35	19
3	49	85	81	231	121	281	250	431	225
4	251	591	637	2251	1271	3231	3086	6267	3475
5	1393	4403	5649	23899	15233	42099	44674	102555	61521

REFERENCES

[1] J. L. GROSS AND T. W. TUCKER, *Generating all graph coverings by permutation voltage assignments*, Discrete Math., 18 (1977), pp. 273–283.
[2] J. L. GROSS AND T. W. TUCKER, *Topological Graph Theory,* John Wiley, New York, 1987.
[3] M. HOFMEISTER, *Counting double covers of graphs*, J. Graph Theory, 12 (1988), pp. 437–444.
[4] M. HOFMEISTER, *Isomorphisms and automorphisms of coverings*, Discrete Math., 98 (1991), pp. 175–183.
[5] M. HOFMEISTER, *Graph covering projections arising from finite vector space over finite fields*, Discrete Math., 143 (1995), pp. 87–97.
[6] M. HOFMEISTER, *Enumeration of concrete regular covering projections*, SIAM J. Discrete Math., 8 (1995), pp. 51–61.
[7] S. HONG AND J. H. KWAK, *Regular fourfold coverings with respect to the identity automorphism*, J. Graph Theory, 15 (1993), pp. 621–627.
[8] S. HONG, J. H. KWAK, AND J. LEE, *Regular graph coverings whose covering transformation groups have the isomorphism extension property*, Discrete Math., 148 (1996), pp. 85–105.
[9] J. H. KWAK AND J. LEE, *Isomorphism classes of graph bundles*, Canad. J. Math., XLII (1990), pp. 747–761.
[10] J. H. KWAK AND J. LEE, *Counting some finite-fold coverings of a graph*, Graphs Combin., 8 (1992), pp. 277–285.
[11] J. H. KWAK AND J. LEE, *Isomorphism classes of cycle permutation graphs*, Discrete Math., 105 (1992), pp. 131–142.
[12] J. H. KWAK AND J. LEE, *Enumeration of graph coverings and its applications*, in Graph Theory, Combinatorics, Algorithms, and Applications: Proc. 7th Quadrennial International Conference on the Theory and Applications of Graphs, Y. Alavi and A. Schwenk, eds., John Wiley, New York, 1995, pp. 649–659.
[13] J. H. KWAK AND J. LEE, *Enumeration of connected graph coverings*, J. Graph Theory, 23 (1996), pp. 105–109.
[14] I. SATO, *Isomorphisms of some coverings*, Discrete Math., 128 (1994), pp. 317–326.
[15] M. SUZUKI, *Group Theory* I, Springer-Verlag, New York, 1982.
[16] M. SUZUKI, *Group Theory* II, Springer-Verlag, New York, 1986.
[17] V. D. TONCHEV, *Combinatorial Configurations Designs, Codes, Graphs*, English version, John Wiley, New York, 1988.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 286–292, May 1998

008

A NOTE ON COUNTING CONNECTED GRAPH COVERING PROJECTIONS*

MICHAEL HOFMEISTER†

Abstract. During the last decade, a lot of progress has been made in the enumerative branch of topological graph theory. Enumeration formulas were developed for a large class of graph covering projections. The purpose of this paper is to count graph covering projections of graphs such that the corresponding covering space is a connected graph. The main tool of the enumeration is Pólya's theorem.

Key words. graph covering, enumeration, Pólya's theorem

AMS subject classifications. 05C10, 05C30, 57M10

PII. S0895480195293873

1. Introduction. In this paper, we consider simple undirected graphs. As usual, the vertex set and the edge set of the graph G are denoted by $V(G)$ and $E(G)$, respectively. An r-to-one graph epimorphism $p : H \to G$ which sends the neighbors of each vertex $x \in V(H)$ bijectively to the neighbors of $p(x) \in V(G)$ is called an *r-fold covering projection* of G. The graph H is the *covering graph*, and the graph G is the *base graph* of p. Topologically speaking, p is a local homeomorphism. The *fibers* of the r-fold covering projection $p : H \to G$ are the sets $p^{-1}(v)$ $(v \in V(G))$. The number r is called the *degree* of p and will be denoted by $\deg(p)$. For an introduction into the field of topological graph theory see, e.g., the famous textbook [2].

Now let $\Gamma \leq \mathrm{Aut}(G)$ be a group of automorphisms of G. There is a natural kind of *isomorphism with respect to* Γ (or Γ-*isomorphism*, for short) between covering projections of G, given by a commutative diagram

(1.1)
$$\begin{array}{ccc} & \overset{\psi}{} & \\ H & \longrightarrow & \tilde{H} \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle \tilde{p}} \\ G & \longrightarrow & G \\ & {\scriptstyle \gamma} & \end{array}$$

with an isomorphism ψ and $\gamma \in \Gamma$.

The interested reader can see much progress in the enumerative branch of topological graph theory during the last decade. Some milestones are the enumeration of double covers of graphs [3], the enumeration of covering projections of labeled graphs (which was established in [4] and [10] independently), and the enumeration of graph coverings with certain regularity properties; examples are [7], [8], and [16]. Concrete and regular concrete graph covering projections are counted in [5] and [6], respectively. Some further interesting papers are [11] and [12].

*Received by the editors September 11, 1995; accepted for publication (in revised form) March 10, 1997.

http://www.siam.org/journals/sidma/11-2/29387.html

†Siemens AG, Corporate Technology, Department ZT SE 4, D-81730 Munich, Germany (Michael.Hofmeister@mchp.siemens.de).

All counting formulas which were obtained so far have no restrictions of connectivity for the covering graph. However, the following question arises immediately: Given the number of covering projections of G up to Γ-isomorphism, how many of them have connected covering graphs? The purpose of this paper is to solve this problem. For this, we will make use of Pólya's enumeration theorem [14] (it should be remembered that it was anticipated in [15]).

It is clear that any covering graph of G is not connected if G is not. Hence, we make the general assumption that the base graph G is connected.

2. Pólya's enumeration theorem. The main tool for enumeration will be Pólya's enumeration theorem. This theorem can be formulated in different and more or less abstract ways. What we need is the so-called *power series formulation*. Let $X = \{1, \dots, n\}$ $(n \in \mathbb{N})$, and let Y be countable. Let Φ be a group acting on X. Clearly, Φ acts on the set of functions Y^X via

$$\varphi(f) = f \circ \varphi^{-1} ,$$

where $f : X \to Y$ and $\varphi \in \Phi$. The orbit of f is denoted by $[f]$.

Now, let $w : Y \to \mathbb{N}_0$ be a *weight function*, i.e., a function with the property that $|w^{-1}(k)| < \infty$ for all $k \in \mathbb{N}_0$. Set $c_k = |w^{-1}(k)|$, and let

$$c(x) = \sum_{k=0}^{\infty} c_k x^k$$

be the *figure counting series*.

The weight of a function $f : X \to Y$ is given by

$$w(f) = \sum_{x \in X} w(f(x)) ,$$

and it is straightforward to see that w is constant on the corresponding orbits $[f]$; hence we may define $w([f]) = w(f)$. Now let C_k be the number of function orbits of weight k. Then the series

$$C(x) = \sum_{k=0}^{\infty} C_k x^k$$

is called the *function counting series*.

Next we define the cycle index of the group Φ acting on X. For $\varphi \in \Phi$, let $(\lambda_1(\varphi), \dots, \lambda_n(\varphi))$ be the *cycle type* of φ, i.e., $\lambda_i(\varphi)$ is the number of cycles of the permutation φ of length i. The *cycle index* of the group Φ acting on X is the polynomial

$$Z(\Phi, s_1, \dots, s_n) = \frac{1}{|\Phi|} \sum_{\varphi \in \Phi} \prod_{i=1}^{n} s_i^{\lambda_i(\varphi)} .$$

For abbreviation, we set, for a power series $q(x)$,

$$Z(\Phi, q(x)) = Z(\Phi, q(x), q(x^2), \dots, q(x^n)) .$$

Pólya's enumeration theorem shows how to obtain the function counting series $C(x)$ from the figure counting series $c(x)$.

THEOREM 2.1. *The function counting series C(x) is determined by substituting the figure counting series c(x) into the cycle index of* Φ:

$$C(x) = Z(\Phi, c(x)).$$

3. A recursion formula for connected covering projections. Let $\mathcal{C}$ be the set of all covering projections of G up to Γ-isomorphism, and let $\overline{\mathcal{C}}$ be the set of all covering projections of G in $\mathcal{C}$ such that the covering graph is connected. Then the degree function deg is a weight function on $\overline{\mathcal{C}}$ (as well as on $\mathcal{C}$). Let c_r be the number of covering projections $p \in \overline{\mathcal{C}}$ of degree r. The figure counting series for the considered problem is the generating function $c(x)$ of the numbers c_r.

Moreover, for $k \in \mathbb{N}$, set $X_k = \{1, \ldots, k\}$. Every function

$$f : X_k \longrightarrow \overline{\mathcal{C}}$$

may be understood as a covering projection p of G with exactly k components. The degree of p is obtained by summing up the degrees of them. Since Γ-isomorphism describes covering projections up to the ordering of the components, we consider the action of the symmetric group S_k on the set X_k. Obviously, the degree function is constant on the orbits $[f]$. By $c_r^{(k)}$ we denote the number of function orbits with degree r, which is exactly the number of r-fold covering projections of G with respect to Γ-isomorphism such that the covering graph consists of exactly k components. The corresponding function counting series is

$$c^{(k)}(x) = \sum_{r=1}^{\infty} c_r^{(k)} x^r \ .$$

LEMMA 3.1. *For every* $k \in \mathbb{N}$,

$$c^{(k)}(x) = Z(S_k, c(x)) \ .$$

In order to prove this lemma, just apply Pólya's enumeration theorem to the described situation.

Let C_r be the number of projections $p \in \mathcal{C}$ of degree r, and let

$$C(x) = \sum_{r=1}^{\infty} C_r x^r$$

be the corresponding counting series. Obviously,

$$C_r = \sum_{k=1}^{r} c_r^{(k)} \ .$$

A short calculation leads to

$$C(x) = \sum_{k=1}^{\infty} c^{(k)}(x) \ .$$

For any power series $q(x)$, we set

$$Z(S_\infty, q(x)) = \sum_{k=0}^{\infty} Z(S_k, q(x)) \ ,$$

where $Z(S_0, q(x))$ is defined to be 1. Using Lemma 3.1, we obtain the following.

LEMMA 3.2. *The counting series for arbitrary and connected graph covering projections of the graph G are related by*

$$1 + C(x) = Z(S_\infty, c(x)) .$$

In order to finish the enumeration, we will follow the approach used by Cadogan [1] for counting connected graphs up to isomorphism by their order.

LEMMA 3.3. *For every power series $q(x)$,*

$$Z(S_\infty, q(x)) = \exp \sum_{k=1}^{\infty} \frac{q(x^k)}{k} .$$

An elegant proof of this well-known identity can be found in [13].

Now define the power series $a(x)$ by setting

$$a(x) = \sum_{r=1}^{\infty} a_r x^r = \log(1 + C(x)) . \tag{3.1}$$

Using $a(x)$, we can formulate a formula for the numbers c_r.

THEOREM 3.4.

(i) *The numbers a_r can be computed by the recursion*

$$r a_r = r C_r - \sum_{k=1}^{r-1} k a_k C_{r-k} .$$

(ii) *The numbers c_r of covering projections of G such that the covering graph is connected can be computed by*

$$c_r = \sum_{d|r} \frac{\mu(d)}{d} a_{r/d} ,$$

where μ is the usual number theoretic Möbius function.

Proof. In order to prove (i), consider the first derivative of equation (3.1):

$$C'(x) = a'(x)\, e^{a(x)} = a'(x)(1 + C(x)) .$$

Comparing coefficients of both sides leads to the recursion formula for the numbers a_r.

From Lemmas 3.2 and 3.3 we obtain

$$\sum_{r=1}^{\infty} a_r x^r = \sum_{k=1}^{\infty} \frac{c(x^k)}{k} .$$

Considering the coefficients leads to

$$r a_r = \sum_{d|r} d c_d .$$

Now the formula of (ii) can be obtained using Möbius inversion. □

4. Connected covering projections of labeled graphs. As an example, we apply Theorem 3.4 to labeled graphs. A graph G is called *labeled* if it is considered together with the trivial automorphism group. In this case, the diagram (1.1) reduces to a triangle

$$\begin{array}{ccc} & \overset{\psi}{} & \\ H & \longrightarrow & \tilde{H} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \tilde{p}} & \\ & G & \end{array} \qquad (4.1)$$

As noticed in the introduction, the enumeration of r-fold covering projections of labeled graphs was done in [4] and [10]. It turned out that the numbers C_r only depend on the Betti number of G, i.e., the number $\beta(G) = m - n + 1$, where m is the number of edges and n is the number of vertices of G.

In order to present the formula given in [4] we need a little bit more terminology.

Let R be the ring of rational polynominals in the variables $s_1, \dots, s_r$. The *cap* product on R, introduced by Redfield [15], is first defined for sequences $s_1^{i_1}\ s_2^{i_2} \dots s_r^{i_r}$, $s_1^{j_1}\ s_2^{j_2} \dots s_r^{j_r}, \dots$ of $q \geq 2$ monomials in R by

$$(s_1^{i_1} \dots s_r^{i_r}) \cap (s_1^{j_1} \dots s_r^{j_r}) \cap \dots = \left(\prod_{k=1}^{r} k^{i_k} i_k! \right)^{q-1}$$

if $i_k = j_k = \dots$ for all k; otherwise it is 0. Then the cap product is linearly extended to arbitrary polynomials in these variables.

Now let $(\lambda) = (\lambda_1, \dots, \lambda_r)$ be a partition of r, i.e.,

$$r = \sum_{i=1}^{r} i\lambda_i.$$

The *partition polynominal* $P(s_1, \dots, s_r)$ is the generating function of the partitions of r:

$$P(s_1, \dots, s_r) = \sum_{(\lambda)} s_1^{\lambda_1} \cdots s_r^{\lambda_r}.$$

THEOREM 4.1. *The number of isomorphism classes of r-fold covering projections of G with respect to the trivial automorphism is*

$$C_r = P(s_1, \dots, s_r)_{\cap}^{\beta(G)}\ .$$

The powers are to be understood with respect to the cap product, which is indicated by the $\cap$-index.

Using Theorems 3.4 and 4.1, we computed for $\beta(G) \leq 10$ and $r \leq 20$ the numbers C_r (extending Table 1 of [4]), ra_r, and c_r. Tables 4.1, 4.2, and 4.3 contain part of our results.[1] For the computations, the comfortable data structures of SYMMETRICA [9] were used.

[1] The reader who is interested in the full tables should send e-mail to Michael.Hofmeister@mchp.siemens.de.

TABLE 4.1
Numbers C_r for labeled graphs.

$\beta(G)\backslash^r$	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1
1	1	2	3	5	7	11	15
2	1	4	11	43	161	901	5579
3	1	8	49	681	14721	524137	25.471105
4	1	16	251	14491	1.730861	373.486525	128038.522439
5	1	32	1393	336465	207.388305	268749.463729	645.244638.648481
6	1	64	8051	7.997683	24883.501301	193.492277.719861	3.252016.862827.895399
7	1	128	47449	191.374041	2.985987.361161	139314.094050.615817	16390.161154.343271.867025
8	1	256	282251	4588.603531	358.318118.583341	100.306131.218514.392365	82.606411.299779.452709.715959

TABLE 4.2
Numbers ra_r for labeled graphs.

$\beta(G)\backslash^r$	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1
1	1	3	4	7	6	12	8
2	1	7	22	111	486	3772	29142
3	1	15	124	2431	68766	3.025596	173.773496
4	1	31	706	56511	8.564226	2229.093460	893451.977874
5	1	63	4084	1.338367	1035.048246	1.611184.631772	4514.783110.968488
6	1	127	23962	31.950591	124375.002186	1160.801154.354052	22.762752.177283.700562
7	1	255	141964	765.274111	14.928949.886766	835866.495874.930476	114730.150164.000899.271416
8	1	511	845986	18353.155071	1791.567290.355426	601.834630.143712.918420	578.244176.306890.931094.903234

TABLE 4.3
Numbers c_r for labeled graphs.

$\beta(G)\backslash^r$	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	1	3	7	26	97	624	4163
3	1	7	41	604	13753	504243	24.824785
4	1	15	235	14120	1.712845	371.515454	127635.996839
5	1	31	1361	334576	207.009649	268530.771271	644.969015.852641
6	1	63	7987	7.987616	24875.000437	193.466859.054994	3.251821.739611.957223
7	1	127	47321	191.318464	2.985789.977353	139311.082645.798043	16390.021452.000128.467345
8	1	255	281995	4588.288640	358.313458.071085	100.305771.690618.678654	82.606310.900984.418727.843319

Note added in proof. After the reviewing process of this paper had been completed, I obtained knowledge of the fact that J.H. Kwak and J. Lee counted connected covering projections of labeled graphs (*Enumeration of connected graph coverings*, J. Graph Theory, 23 (1996), pp. 105–109). However, they used elementary counting methods which do not apply for base graphs with nontrivial automorphism groups.

REFERENCES

[1] C. C. CADOGAN, *The Möbius function and connected graphs*, J. Combin. Theory Ser. B, 11 (1971), pp. 193–200.

[2] J. L. GROSS AND T. W. TUCKER, *Topological Graph Theory*, Wiley Interscience Series in Discrete Mathematics and Optimization, John Wiley and Sons, New York, 1987.

[3] M. HOFMEISTER, *Counting double covers of graphs*, J. Graph Theory, 12 (1988), pp. 437–444.

[4] M. HOFMEISTER, *Isomorphisms and automorphisms of graph coverings*, Discrete Math., 98 (1991), pp. 175–183.

[5] M. HOFMEISTER, *Concrete graph covering projections*, Ars Combin., 32 (1991), pp. 121–127.

[6] M. HOFMEISTER, *Enumeration of concrete regular covering projections*, SIAM J. Discrete Math., 8 (1995), pp. 51–61.

[7] M. Hofmeister, *Graph covering projections arising from finite vector spaces over finite fields*, Discrete Math., 143 (1995), pp. 87–97.
[8] S. Hong and J. H. Kwak, *Regular fourfold coverings with respect to the identity isomorphism*, J. Graph Theory, 17 (1993), pp. 621–627.
[9] A. Kerber and A. Kohnert, *SYMMETRICA*, Program Documentation, Universität Bayreuth, Lehrstuhl II für Mathematik, 1994.
[10] J. H. Kwak and J. Lee, *Isomorphism classes of graph bundles*, Canad. J. Math., 42 (1990), pp. 747–761.
[11] J. H. Kwak and J. Lee, *Counting some finite-fold coverings of a graph*, Graphs Combin., 8 (1992), pp. 277–285.
[12] J. H. Kwak and J. Lee, *Isomorphism classes of cycle permutation graphs*, Discrete Math., 105 (1992), pp. 131–142.
[13] L. Lovasz, *Combinatorial Problems and Exercises*, 2nd ed., North-Holland, Amsterdam, 1993.
[14] G. Pólya, *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen*, Acta Math., 68 (1937), pp. 145–254.
[15] J. H. Redfield, *The theory of group-reduced distributions*, Amer. J. Math., 49 (1927), pp. 433–455.
[16] I. Sato, *Isomorphisms of some graph coverings*, Discrete Math., 128 (1994), pp. 317–326.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 293–304, May 1998

009

A GENERALIZED DISTANCE IN GRAPHS AND CENTERED PARTITIONS*

CRISTIAN LENART†

Abstract. This paper is concerned with a new distance in undirected graphs with weighted edges, which gives new insights into the structure of all minimum spanning trees of a graph. This distance is a generalized one, in the sense that it takes values in a certain Heyting semigroup. More precisely, it associates with each pair of distinct vertices in a connected component of a graph the set of all paths joining them in the minimum spanning trees of that component. A partial order and an addition of these sets of paths are defined. We show how general algorithms for path algebra problems can be used to compute the generalized distance. Some theoretical problems concerning this distance are formulated. The main application of our generalized distance is related to recent clustering procedures. Given a connected graph with weighted edges and certain vertices labeled as *centers*, we define a centered forest to be a spanning forest with exactly one center in each tree component. A partition of the vertices determined by a minimum centered forest will be called a centered partition. These partitions are characterized in terms of the generalized distance, and some corollaries are derived.

Key words. chain distance, Heyting semigroup, generalized distance, semiring, clustering, minimum centered forest, centered partition

AMS subject classifications. 05C12, 62H30

PII. S089548019426303X

1. Introduction.

Several distances in undirected graphs with weighted edges have been defined and studied. For instance, the minimum length distance between two vertices is the smallest length of an elementary path connecting those vertices. By the length of a path, we mean the sum of weights on its edges. If we replace the sum of weights by the maximum weight in the definition of the minimum length distance, we obtain the chain distance. In this paper we define a distance with values in a certain Heyting semigroup (see Definition 2.7). These generalized distances were investigated by Jawhari, Pouzet, and Misane [7]; however, no examples of such distances in undirected graphs, other than the classical ones, were considered in their paper.

In section 2 we construct the Heyting semigroup which we need for the definition of the generalized distance. This construction is based on the definition of a partial order of multisets which generalizes the lexicographic order. In section 3 we define our generalized distance in a graph and prove that it can be expressed in terms of some easily comprehensible graph-theoretic notions; namely, it associates with each pair of distinct vertices in a connected component the set of all paths joining them in the minimum spanning trees (MSTs) of that component. The results of section 2 enable us to "add" and compare such sets of paths. We show that the general algorithms for path algebra problems given in Gondran and Minoux [6] and Pan and Reif [10] can be used to compute the generalized distance. Some theoretical problems related to our distance are formulated.

*Received by the editors February 14, 1994; accepted for publication (in revised form) March 19, 1997.

http://www.siam.org/journals/sidma/11-2/26303.html

†Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (lenart@math.mit.edu).

In section 4 we present an application of our generalized distance to clustering. Several clustering algorithms, such as K-Means (see Tou and Gonzalez [13]), and Fuzzy c-Means (see Bezdek [1]), are able to detect the cores (*centers*) of the clusters, but often misclassify the samples situated close to the borders of the clusters. We have to apply a second clustering algorithm to these remaining samples in order to associate them more accurately with the centers. Such two step clustering procedures were recently proposed in Lenart [9] and Postaire, Zhang, and Lecocq-Botte [11]. The first paper uses the Fuzzy c-Means algorithm to detect the centers of the clusters, while the second one uses morphological transformations. In both papers, a graph-theoretical method similar to the single-linkage method (see Rohlf [12]) is used in the second step. This method, which is investigated in section 4, is well suited for detecting the irregularities of the borders of clusters. The obtained partitions of the set of samples will be called centered partitions. Single-linkage clusters can be expressed in terms of the chain distance; as centered partitions are similar to partitions into single-linkage clusters, it is natural to try to characterize them in metrical terms. We present such a characterization using our generalized distance. Simple examples show that we cannot formulate similar statements in terms of other distances, such as the chain distance. The metrical characterization of centered partitions enables us to give a short, conceptual proof for the maximum split property (see also Delattre and Hansen [4]).

2. Generalized lexicographic orders. Consider a finite set X. A *multiset* with elements from X is a function $A\colon X \to \mathbb{N} \cup \{0, \infty\}$; each image $A(x)$ represents the number of occurrences of x in A. We will call a multiset empty, written $A = \emptyset$, if $A(x) = 0$ for all $x \in X$, and *nonsingular* if $A(x) \neq \infty$ for all $x \in X$. A subset of X can be regarded as a multiset by identifying it with its characteristic function. Let $\mathcal{S}(X)$ denote the set of all subsets of X, $\mathcal{M}(X)$ the set of nonsingular multisets, and $\mathcal{M}_\infty(X)$ the set of all multisets with elements from X.

By virtue of the fact that multisets are mere generalizations of sets, it is natural to make use of set-theoretic notations whenever confusion is not likely. Thus, we will write $x \in A$ to indicate $A(x) \neq 0$ and write $A \subseteq B$ to indicate $A(x) \leq B(x)$ for all $x \in X$; as usual, $A \subset B$ means that $A \subseteq B$ and $A \neq B$. Set difference can be extended naturally to multisets:

$$(A \setminus B)(x) := \max\{A(x) - B(x), 0\}, \ \ \forall x \in X\,. \tag{2.1}$$

The operations called multisum and multiproduct are simply addition and multiplication of multisets as functions; we will write them as $+$ and juxtaposition, respectively.

Throughout this paper, we will write $|V|$ for the cardinality of a given set V, and $[n]$ for the set of integers $\{1, 2, \ldots, n\}$.

Consider a *preferential arrangement* of X (i.e., a partition with a linear order on its blocks) $\sigma(X_1, ..., X_s)$. We can associate with σ a partial order $\preceq_\sigma$ on $\mathcal{M}(X)$ by identifying this set with the ground set of the *lexicographic product* $\bigotimes_{i=1}^{s}(\mathcal{M}(X_i), \subseteq)$ (see, e.g., Davey and Priestley [3]). More explicitly, we have

$$A \prec_\sigma B \iff \exists k \in [s] \,:\, \forall i \in [k-1] \ AX_i = BX_i, \quad \text{and } \ AX_k \subset BX_k\,. \tag{2.2}$$

We will call $\preceq_\sigma$ a *generalized lexicographic order*.

For the rest of this section, we fix a preferential arrangement σ of X and let $\preceq = \preceq_\sigma$. Clearly, $(\mathcal{M}(X), +)$ is a commutative semigroup having the empty set as identity.

PROPOSITION 2.3. *$(\mathcal{M}(X), +, \preceq)$ is a commutative ordered semigroup with identity $\emptyset$.*

Proof. Consider A, B, C in $\mathcal{M}(X)$ such that $A \prec B$, and let k be the smallest positive integer for which $AX_k \subset BX_k$. Then $(A + C)X_k \subset (B + C)X_k$, while $(A + C)X_i = (B + C)X_i$ for all $i \in [k - 1]$. □

It is worth noting that $(\mathcal{S}(X), \cup, \preceq)$ is not an ordered semigroup; this is the main reason that we need to work with multisets and multisum. On the other hand, $(\mathcal{M}(X), \preceq)$ is not a lattice in general, although it is a join-semilattice. To exhibit a counterexample, let $X = \{x_1, x_2, x_3\}$, $\sigma = (\{x_1, x_2\}, \{x_3\})$, $A = \{x_1\}$, and $B = \{x_2\}$. The set of lower bounds of A and B consists of all multisets C with $C(x_1) = C(x_2) = 0$, which does not even have a maximal element. One way to overcome this difficulty is to use *singular multisets*, i.e., multisets in which some elements occur infinitely many often. This approach turns out to be unsatisfactory. Instead, we use a construction which works in a more general setting.

Let $(P, \leq)$ be a poset satisfying the *descending chain condition*; that is, given any sequence $x_1 \geq x_2 \geq \cdots \geq x_n \geq \cdots$ of elements of P, there exists an index k such that $x_k = x_{k+1} = \cdots$. Denote by $\mathcal{S}_a(P)$ the set of all antichains of P, and let $\min\colon \mathcal{S}(P) \to \mathcal{S}_a(P)$ map each subset of P to the set of its minimal elements. Given $Q \subseteq P$, we use the notation $\uparrow Q := \{p \in P \mid \exists q \in Q \ : \ q \leq p\}$. The subset Q is said to be an *up-set* if $\uparrow Q = Q$. The poset $(P^u, \supseteq)$ of all up-sets of P, ordered by reverse inclusion, is known to be a complete and distributive lattice with $\cup$ and $\cap$ as meet and join, respectively (see Davey and Priestley [3, p. 30]). Let us consider the bijection from $\mathcal{S}_a(P)$ to P^u given by $Q \mapsto \uparrow Q$, with inverse $R \mapsto \min R$ (note that the descending chain condition for P is essential). We set $Q \leq R$ in $\mathcal{S}_a(P)$ if and only if $\uparrow Q \supseteq \uparrow R$, thus turning $(\mathcal{S}_a(P), \leq)$ into a complete and distributive lattice. In more explicit terms, we have

$$Q \leq R \iff \forall r \in R, \quad \exists q \in Q \ : \ q \leq r \tag{2.4}$$

for all $Q, R \in \mathcal{S}_a(P)$. Given $Q_i \in \mathcal{S}_a(P)$ for $i \in I$, their meet is specified by

$$\bigwedge_{i \in I} Q_i = \min \bigcup_{i \in I} Q_i \,. \tag{2.5}$$

Clearly, the least element of $\mathcal{S}_a(P)$ is $\min P$, while the greatest element is $\emptyset$.

If P is also equipped with an operation $\circ$ such that $(P, \circ, \leq)$ is an ordered semigroup, then we can extend this operation to $\mathcal{S}_a(P)$ by setting

$$Q \circ R := \min \{q \circ r \mid q \in Q, \, r \in R\} \tag{2.6}$$

for all $Q, R \in \mathcal{S}_a(P)$. It is easy to check that $(\mathcal{S}_a(P), \circ, \leq)$ is also an ordered semigroup. If $(P, \circ)$ has identity e, then $(\mathcal{S}_a(P), \circ)$ has identity $\{e\}$. Moreover, commutativity of $(P, \circ)$ is equivalent to commutativity of $(\mathcal{S}_a(P), \circ)$.

We now return to the ordered semigroup $(\mathcal{M}(X), +, \preceq)$, which, as we have seen, is not a lattice. It can be shown by induction with respect to the cardinality of X that all antichains of $(\mathcal{M}(X), \preceq)$ are finite. Let us recall from Jawhari, Pouzet, and Misane [7] the definition of a *Heyting semigroup*.

DEFINITION 2.7. *$(H, +, \leq)$ is a Heyting semigroup if*

1. *$(H, \leq)$ is a complete lattice with least element 0;*
2. *$(H, +, \leq)$ is an ordered semigroup with identity 0;*
3. *$(\bigwedge_{i \in I} x_i) + (\bigwedge_{j \in J} y_j) = \bigwedge_{i \in I, j \in J} (x_i + y_j)$ for all $x_i, y_j \in H$.*

PROPOSITION 2.8. *$(\mathcal{S}_a(\mathcal{M}(X)), +, \preceq)$ is a commutative Heyting semigroup with identity $\{\emptyset\}$, least element $\{\emptyset\}$, and greatest element $\emptyset$.*

Proof. We will first prove that the poset $(\mathcal{M}(X), \preceq)$ satisfies the descending chain condition. Consider an infinite decreasing sequence $(A_n)_{n\in\mathbb{N}}$ in $\mathcal{M}(X)$. Since A_1X_1 has only a finite number of sub-multisets, we can find $k_1 \in \mathbb{N}$ such that $A_{k_1}X_1 = A_{k_1+1}X_1 = \cdots$. Applying the same argument successively to $X_2, ..., X_s$, we find $k_1 \leq k_2 \leq \cdots \leq k_s$ such that $A_{k_i}X_i = A_{k_i+1}X_i = \cdots$ for all $i \in [s]$. But $A_k = \sum_{i=1}^{s} A_k X_i$ for all k, whence $A_{k_s} = A_{k_s+1} = \cdots$. Conditions (1) and (2) in the definition of a Heyting semigroup now follow from the above discussion about the set of antichains of a given poset. Condition (3) follows from (2.5) and (2.6). □

3. A generalized distance in undirected graphs with weighted edges. Consider $G = (V, E)$ an undirected simple graph (no loops or parallel edges), and a weight function $\rho: E \to [0, \infty)$ with image $\operatorname{Im}\rho = \{r_1, \ldots, r_s\}$; we insist that $r_1 > r_2 > \cdots > r_s$. Given an edge $\{x, y\}$, we will write $\rho(x, y)$, instead of $\rho(\{x, y\})$, for convenience. The triple (V, E, ρ) is known as a *weighted graph*. Consider the preferential arrangement $(\rho^{-1}(r_1), \rho^{-1}(r_2), \ldots, \rho^{-1}(r_s))$ of E and the corresponding generalized lexicographic order $\preceq$ on $\mathcal{M}(E)$. The defining relation (2.2) becomes

$$E_1 \prec E_2 \iff \sup \rho(E_1 \setminus E_2) < \sup \rho(E_2 \setminus E_1) \tag{3.1}$$

for all $E_1, E_2 \in \mathcal{M}(E)$ (as usual, $\sup \emptyset = -\infty$). According to the results in section 2, we have the commutative ordered semigroup $(\mathcal{M}(E), +, \preceq)$ and the commutative Heyting semigroup $(\mathcal{S}_a(\mathcal{M}(E)), +, \preceq)$.

Let us denote by $\mathcal{P}(x, y)$ the set of all x, y-connecting elementary paths, where a given elementary path is identified with its set of edges; thus $\mathcal{P}(x, y) \subseteq \mathcal{S}(\mathcal{S}(E))$. We shall not attempt to distinguish notationally between a path and its set of edges, since in those cases where it matters, we have taken care to ensure that the context is clear. Given a path p, we denote by $\rho(p)$ the multiset of its edge weights. We now define a map $d: V \times V \to \mathcal{S}_a(\mathcal{M}(E))$ by

$$d(x, y) := \begin{cases} \{\emptyset\} & \text{if } x = y, \\ \min \mathcal{P}(x, y) & \text{otherwise.} \end{cases} \tag{3.2}$$

Hence, $d(x, y)$ is the set of x, y-connecting elementary paths which are minimal with respect to $\preceq$. Clearly, d takes values in $\mathcal{S}_a(\mathcal{S}(E))$.

Let us now recall from Jawhari, Pouzet, and Misane [7] the definition of a *generalized distance*.

DEFINITION 3.3. *Let $(H, +, \leq)$ be a Heyting semigroup with least element 0. The function $d: X \times X \to H$ is a generalized distance on X if it satisfies*

1. *$d(x, y) = 0$ if and only if $x = y$;*
2. *$d(x, y) = d(y, x)$ for all $x, y \in X$;*
3. *$d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in X$.*

In this case, (X, d) is called a metric space over H (or generalized metric space).

We are now able to formulate.

PROPOSITION 3.4. *The map d defined by (3.2) is a generalized distance on V.*

Proof. It is straightforward that d is symmetric and that $d(x, y) = \{\emptyset\}$ if and only if $x = y$. Now let $x, y, z \in V$. If at least two of them coincide or lie in different connected components of G, the triangle inequality is obvious. Otherwise, let $p_1 + p_2$ be a typical element of $d(x, z) + d(z, y)$, where $p_1 \in d(x, z)$ and $p_2 \in d(z, y)$. Clearly, $\mathcal{P}(x, y)$ contains a minimal element $\preceq p_1 + p_2$, so the triangle inequality follows. □

Let us observe that if the vertices x and y lie in different connected components of G, then $d(x,y)=\emptyset$; this is natural, because $\emptyset$ is the greatest element of $\mathcal{S}_a(\mathcal{M}(E))$. If all the edge weights are different, then $d(x,y)$ contains at most one path, and $(\mathcal{M}_\infty(E),+,\preceq)$ is a commutative Heyting semigroup. Hence, in this case, we can let d take values in $\mathcal{M}_\infty(E)$. We need to work with $\mathcal{S}_a(\mathcal{M}(E))$ only in the degenerate case, when several edge weights are equal. If *all* the edge weights are equal (or if we simply discard ρ), then $d(x,y)$ consists of all x,y-connecting elementary paths. In general, $d(x,y)$ is characterized by Theorem 3.6, which will be proved using the following lemma.

LEMMA 3.5. *Let $p,p'\in\mathcal{P}(x,y)$, $x\neq y$ be two elementary paths. Every edge e of p not in p' belongs to a subpath of p with end-vertices connected by a subpath of p' with no edges in p.*

Proof. Write p' as a concatenation of subpaths $p'=q_1q_1',\dots,q_kq_k'q_{k+1}$, where q_i', $i\in[k]$ are all the maximal subpaths of p' with no edges in p; note that q_1 or q_{k+1} may contain a single vertex. As p is elementary, every q_i is a subpath of p. Let $j\in[k]$ be smallest possible such that q_{j+1} lies on the subpath of p from e to y. We deduce that the end-vertices of q_j' are connected by a subpath of p containing e. □

The proof of Theorem 3.6 is based on Kruskal's algorithm for the MST of a weighted connected graph. Recall that this algorithm considers the edges of the graph in increasing order of weight and selects those which do not form a circuit with some edges already selected.

THEOREM 3.6. *If G is connected then, for x and y distinct, $d(x,y)$ consists of all paths connecting x and y in an MST of G.*

Proof. Let $p'\prec p$ be two elementary paths in $\mathcal{P}(x,y)$. Let e be an edge of maximum weight in $p\setminus p'$. According to the lemma, e belongs to a subpath q of p with end-vertices connected by a subpath q' of p' with no edges in p. We have $\max\rho(q')<\rho(e)$. Kruskal's algorithm is never able to select all the edges of q, because when edges of weight $\rho(e)$ are considered, a path connecting the end-vertices of q has already been selected.

Now let $p\in\mathcal{P}(x,y)$, $x\neq y$ be an elementary path which is not contained in any MST of G. Apply Kruskal's algorithm, always choosing an edge of p if possible, when ties appear. Denote by T the resulting MST and by p' the elementary x,y-connecting path in T. Consider a maximal subpath q' of p' not containing edges of p and the subpath q of p with the same end-vertices as q'. Write q as a concatenation of subpaths $q=q_1'q_1,\dots,q_k'q_kq_{k+1}'$, where q_i, $i\in[k]$, are all the maximal subpaths of q not containing edges of p' (q_1 and q_{k+1} may contain a single vertex). Each edge e in one of the subpaths q_i has its end-vertices connected by a path in T with edge weights not greater than $\rho(e)$. Furthermore, according to the algorithm, all the edges of this path of weight $\rho(e)$ are in p. Now concatenate all these paths, and the paths q_i', $i\in[k+1]$, to obtain a new path q'' connecting the end-vertices of q', and having all edges in T. Removing all cycles from q'', we obtain q'. Hence, all edges in q'' belonging to p (including those from the paths q_i'), were removed. We deduce that $\max\rho(q')<\max\{\max\rho(q_i)\mid i\in[k]\}$. As q' was chosen arbitrarily, we have that $p'\prec p$. □

Proposition 3.9 and the remarks following it address the relation between the distance d and *chain distance* $\delta\colon V\times V\to[0,\infty]$, defined for all $x,y\in V$ by

$$\delta(x,y):=\begin{cases}0 & \text{if } x=y,\\ \inf\{l(p)\mid p\in\mathcal{P}(x,y)\} & \text{otherwise,}\end{cases}\tag{3.7}$$

where $l(p):=\max\rho(p)$ (as usual, $\inf\emptyset=\infty$). It is well known that δ is an ultrametric.

Let us note that l takes the same value on all paths in $d(x,y)$. We also address the relation between the distance d and a certain distance $\widetilde{d}$, which we now define. Consider the Heyting semigroup $(\mathcal{M}_\infty(\operatorname{Im}\rho),+,\leq)$ of multisets of edge weights; here $\leq$ is the lexicographic order for multisets of real numbers, with the largest values being most significant. We define $\widetilde{d}(x,y)$ to be the unique minimal multiset of edge weights in $\{\rho(p)\,|\,p\in\mathcal{P}(x,y)\}$, provided that $\mathcal{P}(x,y)$ is nonempty; if $x=y$, we set $\widetilde{d}(x,y)=\emptyset$, and if x and y lie in different connected components, we define $\widetilde{d}(x,y)$ to be the greatest element of $\mathcal{M}_\infty(\operatorname{Im}\rho)$. The order $\leq$ defines a preorder on the set of paths in $\mathcal{P}(x,y)$ (assumed to be nonempty), in the sense that $p\leq q$ if and only if $\rho(p)\leq\rho(q)$. Let p_0 be a minimal element with respect to this preorder; in other words $\rho(p_0)=\widetilde{d}(x,y)$. We notice an analogy between the minimality of p_0 in $\mathcal{P}(x,y)$ and the minimality of an MST in the set of spanning trees (in the latter case, there is a similar preorder, and it does not matter whether we consider smallest or largest weights most significant in the definition of the lexicographic order for multisets of weights). However, in the MST case we have the stronger property of *Gale-optimality* (see Gale [5] or Lawler [8, p. 277]), which does not appear to have an analogue for paths.

The proof of Proposition 3.9 is based on the following lemma.

LEMMA 3.8.

1. *If $p\prec q$ are paths in $\mathcal{P}(x,y)$, then $\rho(p)<\rho(q)$.*
2. *Any path p_0 in $\mathcal{P}(x,y)$ with $\rho(p_0)=\widetilde{d}(x,y)$ lies in $d(x,y)$.*

Proof. 1. Indeed, by (3.1), there is a number v (in fact $v=\sup\rho(q\setminus p)$) such that the set of edges of p with weights greater or equal to v is strictly contained in the set of edges of q with weights greater or equal to v; but this implies $\rho(p)<\rho(q)$ in lexicographic order.

2. If $p_0\notin d(x,y)$, there is a path p in $d(x,y)$ with $p\prec p_0$, which contradicts the minimality of $\rho(p_0)$ by the first part of the lemma. □

PROPOSITION 3.9. *Given x,y,u,v in V, the following hold:*

1. *If $\delta(x,y)<\delta(u,v)$, then $d(x,y)\prec d(u,v)$.*
2. *If $d(x,y)\prec d(u,v)$, then $\widetilde{d}(x,y)<\widetilde{d}(u,v)$.*

Proof. The first part follows from the fact that l takes the same value on all paths in $d(x,y)$ and $d(u,v)$. For the second part, assume that $d(u,v)\neq\emptyset$, and let p_0 be a path in $\mathcal{P}(u,v)$ for which $\rho(p_0)=\widetilde{d}(u,v)$. We have $p_0\in d(u,v)$ by Lemma 3.8 (2). According to (2.4), there is a path p in $d(x,y)$ such that $p\prec p_0$. Hence $\rho(p)<\rho(p_0)$, by Lemma 3.8 (1). □

The converses do not hold. Indeed, consider the weighted graphs in Figs. 1 and 2. For the graph in Fig. 1 we have $d(x_1,x_3)\prec d(x_1,x_4)$, while $\delta(x_1,x_3)=\delta(x_1,x_4)$. For the graph in Fig. 2 we have $\widetilde{d}(x_1,x_3)<\widetilde{d}(x_1,x_5)$, while $d(x_1,x_3)$ and $d(x_1,x_5)$ are incomparable. Hence, we could say that the "discriminating power" of d lies between that of δ and $\widetilde{d}$. It turns out that this is exactly what we need for our applications in the following section.

A property which does not hold for δ, but holds for d (as well as for the minimum length distance, for which it was first defined), is Bellman's principle. It says that every subpath of a minimal path is also minimal (with respect to a certain order, in the corresponding set $\mathcal{P}(x,y)$). To see that this is not generally true for δ, consider the weighted graph in Fig. 3; take the path $(\{x_1,x_2\},\{x_2,x_4\},\{x_4,x_5\})\in\mathcal{P}(x_1,x_5)$, minimal with respect to δ, and $(\{x_2,x_4\},\{x_4,x_5\})\in\mathcal{P}(x_2,x_5)$, which is not minimal.

PROPOSITION 3.10 (Bellman's principle). *Let p be a path in $d(x,y)$, and let q be a u,v-connecting subpath of p. Then q lies in $d(u,v)$.*

Proof. This statement is immediate using the characterization of $d(x,y)$ given in Theorem 3.6. □

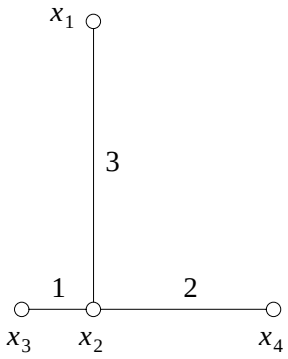


FIG. 1. *A weighted graph with* $\rho(x_2, x_3) < \rho(x_2, x_4) < \rho(x_1, x_2)$.

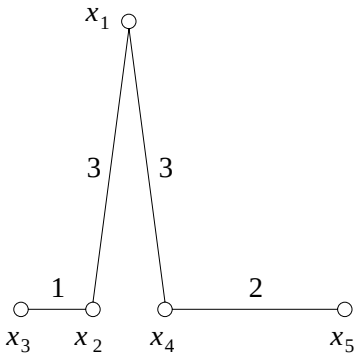


FIG. 2. *A weighted graph with* $\rho(x_2, x_3) < \rho(x_4, x_5) < \rho(x_1, x_2) = \rho(x_1, x_4)$.

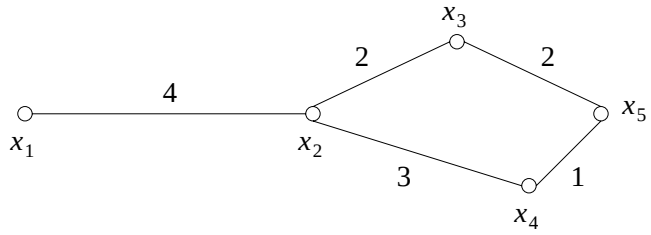


FIG. 3. *A weighted graph with* $\rho(x_4, x_5) < \rho(x_2, x_3) \leq \rho(x_3, x_5) < \rho(x_2, x_4) < \rho(x_1, x_2)$.

In order to give algorithms for the computation of the distance d, we formulate this problem as a path algebra problem. First, note that Proposition 2.8 implies that $(\mathcal{S}_a(\mathcal{M}(E)), \wedge, +)$ is a commutative *semiring* (semirings are generalizations of rings in the sense that subtraction may not be defined, see, e.g., Wongseelashote [14]). We denote by $\oplus$ and $\otimes$ matrix addition and multiplication in the semiring $(\mathcal{S}_a(\mathcal{M}(E)), \wedge, +)$; that is,

$$B \oplus C = (b_{ij} \wedge c_{ij}), \qquad B \otimes C = \left(\bigwedge_{k=1}^{n} b_{ik} + c_{kj} \right),$$

where $B = (b_{ij})$ and $C = (c_{ij})$ are $n \times n$ matrices. Given a weighted graph (V, E, ρ) with $V = \{x_1, x_2, \ldots, x_n\}$, let $A = (a_{ij})$ be the $n \times n$ matrix with elements

$$a_{ij} := \begin{cases} \{\emptyset\} & \text{if } i = j, \\ \{\{e\}\} & \text{if } e = \{x_i, x_j\} \in E, \\ \emptyset & \text{otherwise.} \end{cases}$$

Let X be the row vector of distances from x_1, say, to the vertices in V, and I the vector $(\{\emptyset\}, \emptyset, \ldots, \emptyset)$. We claim that the vector X satisfies the well-known fixpoint equation

$$X = X \otimes A \oplus I\,. \tag{3.11}$$

Indeed, all the entries of X equal to $\{\emptyset\}$ or $\emptyset$ are clearly equal to the corresponding entries of the right-hand side. Now let $x_i \neq x_1$ be a vertex in the same connected

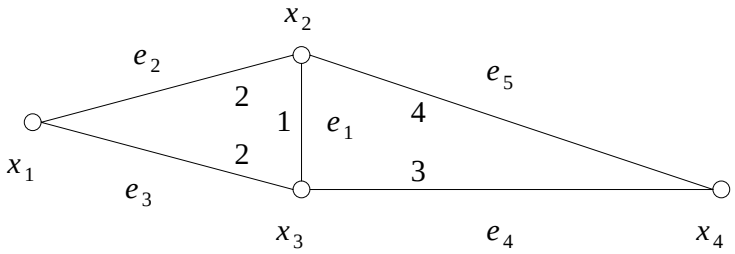


FIG. 4. *A weighted graph with* $\rho(e_1) < \rho(e_2) = \rho(e_3) < \rho(e_4) < \rho(e_5)$.

component as x_1. By definition, the ith entry of $X \otimes A \oplus I$ consists of the minimal elements of a certain set of x_1, x_i-connecting paths containing $d(x_1, x_i)$; hence, this entry coincides with $d(x_1, x_i)$.

According to (3.11), the general algorithms (over semirings) for the *single source shortest path* problem given in Gondran and Minoux [6] and Pan and Reif [10] can be applied. For instance, we can use the generalized Jacobi algorithm (see Gondran and Minoux [6, p. 105]); recall that this algorithm sets

$$X^1 := (\{\emptyset\}, a_{12}, \ldots, a_{1n}),$$

and at each stage k computes

$$X^{k+1} := X^k \otimes A \oplus I$$

until $X^{k+1} = X^k$. We can also apply algorithms for the *all pairs shortest path* problem.

Example 3.12. Consider the weighted graph in Fig. 4. The vector X is determined in the following three steps using the generalized Jacobi algorithm:

$$\begin{aligned}
X^1 &= (\,\{\emptyset\},\ \{\{e_2\}\},\ \{\{e_3\}\},\ \emptyset\,)\\
X^2 &= (\,\{\emptyset\},\ \{\{e_2\},\{e_3,e_1\}\},\ \{\{e_3\},\{e_2,e_1\}\},\ \{\{e_3,e_4\}\}\,)\\
X^3 &= (\,\{\emptyset\},\ \{\{e_2\},\{e_3,e_1\}\},\ \{\{e_3\},\{e_2,e_1\}\},\ \{\{e_2,e_1,e_4\},\{e_3,e_4\}\}\,).
\end{aligned}$$

Similarly to the definition of the distance between two vertices, we can define the distance between two subsets of V. Given U, W such subsets, we denote by $\mathcal{P}(U, W)$ the set of elementary paths with one end-vertex in U and the other in W. The distance between U and W is defined by

$$d(U, W) := \begin{cases} \{\emptyset\} & \text{if } U \cap W \neq \emptyset,\\ \min \mathcal{P}(U, W) & \text{otherwise.} \end{cases} \tag{3.13}$$

The following result is to be expected.

PROPOSITION 3.14. *For* $U, W \subseteq V$*, we have that*

$$d(U, W) = \wedge \{d(x, y) \mid x \in U,\ y \in W\}.$$

Proof. If $U \cap W = \emptyset$, the relation to be proved may be written

$$\min \mathcal{P}(U, W) = \min \bigcup_{x \in U,\, y \in W} d(x, y).$$

But this is equivalent to

$$\min \bigcup_{x \in U,\, y \in W} \mathcal{P}(x, y) = \min \left(\bigcup_{x \in U,\, y \in W} \min \mathcal{P}(x, y) \right),$$

which is obvious. □

Having defined the distance d, we can now construct new graph concepts, following the model of certain concepts defined in terms of classical distances. We give some examples and problems related to them, which could be of theoretical interest.

CONVEXITY. Given $x, y \in V$, the *interval* (with respect to d) $I(x, y)$ is defined to be the set of all vertices z such that $d(x, y) \cap (d(x, z) + d(z, y)) \neq \emptyset$ (requiring equality in the triangle inequality would be too restrictive). Using Theorem 3.6, we deduce that if $x \neq y$ are in the same connected component, $I(x, y)$ consists of all the vertices of the x, y-connecting paths in the MSTs of that component; otherwise, $I(x, y) = \emptyset$. A set $U \subseteq V$ is called *convex* (with respect to d) if $I(x, y) \subseteq U$ for all $x, y \in U$. Clearly, the distance d in an induced subgraph is the restriction of the distance in the whole graph if and only if the subgraph is induced by a convex set of vertices. Such subgraphs could be called *isometric* (with respect to d). It can be shown, for instance, that the sets $\{y \mid \delta(x, y) \leq r\},\ x \in V,\ r \geq 0$, (known as *single-linkage clusters*, see Rohlf [12]) are convex. Does every weighted graph have all intervals convex?

DISTANCE MONOTONE GRAPHS. The *diameter* of a set $U \subseteq V$ is defined by $D(U) := \vee \{d(x, y) \mid x, y \in U\}$. An interval $I(x, y)$ is called *closed* (with respect to d) if $d(z, z') \preceq D(I(x, y))$ for all $z' \in I(x, y)$ implies $z \in I(x, y)$. The interval $I(x, y)$ is called strictly closed (with respect to d) if for each $z \in V \setminus I(x, y)$ there is $z' \in I(x, y)$ such that $d(z, z') \succ D(I(x, y))$. Clearly, a strictly closed interval is closed, but not conversely. A graph is called (strictly) *distance monotone* (with respect to d) if all its intervals are (strictly) closed. What can we say about (strictly) distance monotone graphs? The same problem in the case of the minimum length distance and graphs with all edges of weight 1 is addressed in Burosch, Havel, and Laborde [2].

TREE METRICS. The distance d is called a *tree metric* if it satisfies the 4-point condition, i.e., $d(x_1, x_2) + d(x_3, x_4) \preceq (d(x_1, x_3) + d(x_2, x_4)) \vee (d(x_1, x_4) + d(x_2, x_3))$ for all $x_1, x_2, x_3, x_4 \in V$. In general, d is not a tree metric. Nevertheless, if our weighted graph has a unique MST, then d is a tree metric; moreover, two of the sums of distances in the 4-point condition are equal, and $\succeq$ than the third one. Does the 4-point condition (or the stronger statement before) imply uniqueness of the MST?

We conclude this section by suggesting that the results of Jawhari, Pouzet, and Misane [7] on retraction and fixed-point property for generalized metric spaces, as well as those of Wongseelashote [14] concerning path spaces, might have useful consequences in our setting.

4. Centered partitions. Let us consider a weighted simple graph $G = (V, E, \rho)$ with a distinguished set C of vertices, which will be called *centers*. We assume that G is connected, that there are no edges between centers, and that ρ does not take the value 0. Let $C = \{c_1, ..., c_n\}$, and $|V \setminus C| = m$. We define a *centered forest* of G to be a spanning forest with exactly one center in each tree component. A centered forest which minimizes the sum of weights on its edges will be called a *minimum centered forest* (MCF). A partition of V determined by the components of an MCF will be called a *centered partition*. We say that a centered partition assigns a vertex x to a center c if x and c belong to the same block of that partition.

The terminology in the above paragraph is inspired by a *clustering* problem which can be modeled by the weighted graph G with the distinguished set C of vertices. These vertices (centers) represent the cores of some clusters of samples; they were determined by a previous clustering algorithm (e.g., Fuzzy c-Means, as in Lenart [9], or morphological transformations, as in Postaire, Zhang, and Lecocq-Botte [11]). The vertices in $V \setminus C$ represent the samples situated close to the borders of the clusters; they could not be classified by the previous algorithm. Each pair of vertices, except pairs of centers, determine an edge. The weights are given by a dissimilarity coefficient, which

could be Euclidean distance, if the samples are embedded in $\mathbb{R}^s$. Centered partitions are, in a certain sense, optimal with respect to the objectives of clustering. They are similar to partitions into single-linkage clusters, which are obtained by removing from an MST all the edges with weight greater than a certain threshold.

Let G_0 be the weighted graph obtained from G by adding all edges between centers, and assigning to them the weight 0. Clearly, an MST of G_0 gives rise to an MCF of G by removing the edges of weight 0; moreover, all the MCFs of G arise in this way. Therefore, we can use slight variations of classical algorithms for the MST in order to determine an MCF. For instance, in Kruskal's algorithm we must select the edges of G in the increasing order of their weights, discarding not only those which form cycles, but also those which form paths connecting two centers. In Prim's algorithm, the only step to be modified is initializing the set of vertices incident to selected edges with C, instead of a single vertex set. This version of Prim's algorithm, which will be used in the proof of Lemma 4.2, is described in detail below.

ALGORITHM 4.1.

STEP 1. *Let* $k := 1$; $N^1 := V \setminus C$; $E^1 := \emptyset$; $V_i^1 := \{c_i\}$, *for* $i \in [n]$.

STEP 2. *Let* $L^k := N^k \times (V \setminus N^k)$. *Search for the minimum weight edge* (x^k, y^k) *in* L^k.

STEP 3. *For* $i \in [n]$ *do : if* $y^k \in V_i^k$, *then* $V_i^{k+1} := V_i^k \cup \{x^k\}$, *else* $V_i^{k+1} := V_i^k$.

STEP 4. *Let* $E^{k+1} := E^k \cup \{\{x^k, y^k\}\}$; $N^{k+1} := N^k \setminus \{x^k\}$.

STEP 5. *Let* $k := k + 1$. *If* $N^k \neq \emptyset$, *then go to* (2), *else* STOP.

This algorithm finishes after m interations. The set N^k contains the objects not yet classified, L^k is the set of edges in which the least weight edge is searched for, and E^k is the set of selected edges after $k-1$ iterations. At the kth iteration, exactly one object is added to one of the growing clusters V_i^k, $i \in [n]$. The output centered partition of V has blocks V_i^{m+1}, $i \in [n]$, while E^{m+1} contains the edges of the corresponding MCF. When ties appear at step 2, we obtain the family of centered partitions of V choosing the minimum weight edge in all possible ways.

As we pointed out in the introduction, the analogy between centered partitions and single-linkage clusters (which are expressed in terms of the chain distance) shows that it is natural to characterize the former in metrical terms. The main result of this section (Theorem 4.4) is a characterization of centered partitions using the generalized distance d. The following lemma contains the essential part of the proof.

LEMMA 4.2. *The partition* $\{V_i \mid i \in [n]\}$ *of* V, *satisfying* $c_i \in V_i$ *for all* i, *is a centered partition if and only if*

$$\forall i \in [n], \quad \forall x \in V_i \ : \ d(x, c_i) \cap d\left(x, \bigcup_{j \neq i} V_j \cup \{c_i\}\right) \neq \emptyset. \tag{4.3}$$

Proof. ($\Rightarrow$) Let T_0 be the MST of G_0 which determines the given centered partition, and let F be the corresponding MCF. Consider an arbitrary vertex x in C_i, and the path p connecting c_i and x in F. Note that F is contained in an MST of G, whence, by Theorem 3.6, p lies in $d(x, c_i)$. Now suppose that there is a path $q \prec p$ connecting x to a vertex in $V \setminus V_i$. We can view the path q as a concatenation of paths $q_i e q_j$, where q_i has all vertices in V_i, and $e = \{y, z\}$ is an edge with $y \in V_i$ and $z \in V_j$, $j \neq i$. Since $q \prec p$, then, by (3.1), the edge e' of maximum weight in $p \setminus q$ satisfies $\rho(e) < \rho(e')$. Let p_j be the z, c_j-connecting elementary path in F, and p_{ji} the c_j, c_i-connecting elementary path in T_0. Let q_i' be the x, y-connecting path obtained from q_i by replacing all edges not lying in F with paths in F connecting their end-vertices. Using the fact that $q \prec p$ again, we deduce that the edge e' appears only once in the path pq_i', whence it is contained in the elementary path p_i obtained from

pq_i' by removing all cycles. We have thus constructed the z, y-connecting elementary path $p_j p_{ji} p_i$ in T_0, which contains e'. But this contradicts the fact that T_0 is an MST of G_0, whence p lies in $d(x, \bigcup_{j\neq i} V_j \cup \{c_i\})$.

($\Leftarrow$) Assume that the given partition satisfies (4.3). We will prove that for each $k \in [m]$, we can select $(x^k, y^k) \in L^k$ at Step 2 of Algorithm 4.1 such that $V_i^{k+1} \subseteq V_i$ for all $i \in [n]$. Let $(x, y) \in L^k$ be a minimum weight edge. If $x \in V_i$ and $y \in V_i^k$ for some $i \in [n]$, we set $x^k := x$ and $y^k := y$. Otherwise, suppose that $x \in V_j,\ y \in V_i^k,\ i \neq j$, and choose a path p in $d(x, c_j) \cap d(x, \bigcup_{l\neq j} V_l \cup \{c_j\})$. Since $x \in N^k$ and all the vertices of p lie in V_j (otherwise p has a subpath lying in $d(x, \bigcup_{l\neq j} V_l)$, which contradicts (4.3)), there is an edge $\{u, v\}$ of p with $u \in N^k \cap V_j$ and $v \in V_j^k$. Hence, (u, v) lies in L^k, whence $\rho(u, v) \geq \rho(x, y)$. On the other hand, from the minimality of p it follows that we cannot have $\{\{x, y\}\} \prec p$. Thus, (u, v) is also of minimum weight in L^k. We now set $x^k := u$ and $y^k := v$, which ensures that $V_j^{k+1} \subseteq V_j$. □

THEOREM 4.4. *Consider a partition of the vertices determined by a centered forest* $\{V_i \mid i \in [n]\}$*, with* $c_i \in V_i$ *for all* i*. This partition is* not *a centered partition if and only if the following condition holds*:

$$\exists i \in [n], \qquad \exists x \in V_i \ :\ d(x, V \setminus V_i) \prec d(x, c_i). \tag{4.5}$$

Proof. Consider arbitrary $i \in [n]$ and $x \in V_i$. According to the defining relation (2.4), the condition $d(x, V \setminus V_i) \preceq d(x, c_i)$ is equivalent to

$$\forall p \in d(x, c_i), \quad \exists q \in d(x, V \setminus V_i) \ :\ q \preceq p\,.$$

But equalities cannot hold, whence condition (4.5) is equivalent to the negation of (4.3). □

COROLLARY 4.6. *There is no centered partition which assigns the vertex* x *to the center* c *if and only if* $d(x, C \setminus \{c\}) \prec d(x, c)$.

Proof. The "if" part follows immediately from Theorem 4.4. Now assume that $d(x, C \setminus \{c\}) \not\prec d(x, c)$. According to (2.4) and (3.13), this means that we can find a path $p \in d(x, c) \cap d(x, C)$. Apply Kruskal's algorithm for the MCF, always choosing an edge of p, if possible, when ties appear. Suppose that x is not assigned to c. Consider the first iteration in the algorithm when an end-vertex of an edge in p is assigned to a center c' different from c. Let y be such a vertex, which is closest to c on the path p, and let q denote the y, c'-connecting path. Let $p_1 \in \mathcal{P}(y, c)$ and $p_2 \in \mathcal{P}(x, y)$ such that $p = p_1 + p_2$. Obviously, $p_1 \cap q = \emptyset$. According to the algorithm, we have that $q \prec p_1$, which implies that $q + p_2 \prec p_1 + p_2 = p$, but this contradicts the minimality of p in $d(x, C)$. Hence, x is assigned to c by the algorithm. □

It is not possible to reformulate Theorem 4.4 or Corollary 4.6 in terms of the distances δ or $\widetilde{d}$ mentioned in section 3. Indeed, consider the weighted graph in Fig. 1, and take x_3 and x_4 as centers, obtaining the unique centered partition $\{\{x_1, x_2, x_3\}, \{x_4\}\}$; we have $\delta(x_1, x_3) = \delta(x_1, x_4)$, which means that we cannot specify that x_1 should be assigned to x_3 in terms of δ. Consider the weighted graph in Fig. 2, and take x_3 and x_5 as centers; the partition $\{\{x_2, x_3\}, \{x_1, x_4, x_5\}\}$ is a centered partition, but $\widetilde{d}(x_1, \{x_2, x_3\}) < \widetilde{d}(x_1, x_3) < \widetilde{d}(x_1, x_5)$.

As we mentioned in the introduction, the metrical characterization of centered partitions (Theorem 4.4) is useful for studying certain properties of these partitions. One of these properties is concerned with their *split*, a concept which we now define. The *cocycle* generated by a set $U \subseteq V$, denoted by $c(U)$, is the set of all edges with one end-vertex in U and the other in $V \setminus U$. The split of U, denoted by $s(U)$, is defined by $s(U) := \inf\{\rho(e) \mid e \in c(U)\}$, where, as usual, $\inf \emptyset = \infty$. The split of the partition

$\pi = \{V_1, ..., V_n\}$ of V, denoted $s(\pi)$, is defined by $s(\pi) := \min\{s(V_i) \mid i \in [n]\}$. In Delattre and Hansen [4], it is shown that a partition of the vertices into single-linkage clusters has the maximum split among all partitions with the same number of blocks. We will prove a similar result for centered partitions, using Theorem 4.4.

COROLLARY 4.7. *Centered partitions maximize the split among all partitions determined by centered forests.*

Proof. Consider a centered partition $\pi = \{V_1, ..., V_n\}$. Choose an edge $e = \{x, y\}$ with $x \in V_i$, $y \in V_j$, $i \neq j$, such that $\rho(e) = s(\pi)$. We clearly have $d(x, V \setminus V_i) \preceq \{e\}$, but we cannot have $\{e\} \prec d(x, c_i)$, because this would imply $d(x, V \setminus V_i) \prec d(x, c_i)$, which contradicts the fact that π is a centered partition, by Theorem 4.4. Therefore, according to the defining relation (2.4), there is a path $p_i \in \mathcal{P}(x, c_i)$ with all edge weights (if any) less than or equal to $\rho(e)$. We can find a path $p_j \in \mathcal{P}(y, c_j)$ with the same property. Hence, the concatenation $p_i e p_j$ is a path connecting c_i and c_j with all edge weights less than or equal to $\rho(e)$. But a partition of V determined by a centered forest separates at least a pair of consecutive vertices on this path. The conclusion now follows. □

Acknowledgments. The author is grateful to the referees for suggesting simplifications in the proof of (2.5), and of the "⇒" part of Lemma 4.2.

REFERENCES

[1] J. C. BEZDEK, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
[2] G. BUROSCH, I. HAVEL, AND J. M. LABORDE, *Distance monotone graphs and a new characterization of hypercubes*, Discrete Math., 110 (1992), pp. 9–16.
[3] B. DAVEY AND H. PRIESTLEY, *Introduction to Lattices and Order*, Cambridge Mathematical Textbooks, Cambridge University Press, 1990.
[4] M. DELATTRE AND P. HANSEN, *Bicriterion cluster analysis*, IEEE Trans. Pattern Anal. Machine Intell., PAMI-2 (1980), pp. 277–291.
[5] D. GALE, *Optimal assignments in an ordered set: An application of matroid theory*, J. Combin. Theory, 4 (1968), pp. 176–180.
[6] M. GONDRAN AND M. MINOUX, *Graphs and Algorithms*, Wiley Interscience, New York, 1984.
[7] M. JAWHARI, M. POUZET, AND D. MISANE, *Retracts: Graphs and ordered sets from the metric point of view*, in Combinatorics and Ordered Sets (Arcata, Calif., 1985), Amer. Math. Soc., Contemp. Math. 57, AMS, Providence, RI, 1986, pp. 175–226.
[8] E. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.
[9] C. LENART, *Clustering and Learning in Pattern Recognition*, Ph.D. thesis, Univ. of Cluj-Napoca, Romania, 1992.
[10] V. PAN AND J. REIF, *Fast and efficient solution of path algebra problems*, J. Comput. System Sci., 38 (1989), pp. 494–510.
[11] J.-G. POSTAIRE, R. D. ZHANG, AND C. LECOCQ-BOTTE, *Cluster analysis by binary morphology*, IEEE Trans. Pattern Anal. Machine Intell., PAMI-15 (1993), pp. 170–180.
[12] F. J. ROHLF, *Single-link clustering algorithms*, in Classification, Pattern Recognition and Reduction of Dimensionality, Handbook of Statistics 2, P. R. Krishnaiah and L. N. Kanal, eds., North-Holland, Amsterdam-New York, 1982, pp. 267–284.
[13] J. T. TOU AND R. C. GONZALEZ, *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, 1974.
[14] A. WONGSEELASHOTE, *Semirings and path spaces*, Discrete Math., 26 (1979), pp. 55–78.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 305–317, May 1998

010

CAYLEY DIGRAPHS BASED ON THE DE BRUIJN NETWORKS*

M. ESPONA† AND O. SERRA†

Abstract. A construction of Cayley digraphs associated to arc-colored regular digraphs is presented. The resulting Cayley digraphs, which we call Cayley regular covers, can be seen as a symmetrization of the original digraph. This construction is applied to the de Bruijn digraphs. By using the fact that they are iterated line digraphs of complete symmetric digraphs, valuable information about their Cayley regular covers regarding routings, diameter, hamiltonicity, fault-tolerance properties and degree of symmetry is obtained. In particular, a shortest-path, self-routing algorithm is given for a family of Cayley digraphs which includes the well known butterfly network.

These results can be applied to the design of permutation networks. The Cayley regular covers represent sets of permutations in the original digraph which can be performed without conflict. In particular, a sharply 2-transitive group of permutations on the de Bruijn network is presented which admits a simple shortest-path self-routing algorithm. By using the same construction, a Cayley digraph on the symmetric group on the nodes of the de Bruijn digraph of degree two is obtained. The techniques introduced in this paper can also be extended to other families of iterated line digraphs.

Key words. Cayley digraphs, permutation networks, routing algorithms

AMS subject classifications. 05C20, 05C70, 94C15

PII. S0895480196288644

1. Introduction.

The performance of massive parallel computers relies heavily on the properties of the interconnection network that connects processors and memories or processors among themselves. Several authors have emphasized the importance of symmetry properties of the network for both algorithmic efficiency and good fault-tolerance (see, for instance, [2, 3, 8, 12, 17]). Actually, most of the symmetric networks proposed in the literature and used in commercial machines are based on Cayley graphs. This is the case of the *hypercube* [25], *butterfly* [3, 19], *cube-connected cycles* [23], and *star graphs* [1] among others. Besides, there are a number of networks which have been proposed as possible alternatives to the symmetric ones, such as the *shuffle-exchange* network [27], the *de Bruijn* and *Kautz* networks [5], and their variants.

In [3], Annexstein, Baumslag, and Rosenberg introduced the *group action graphs* which provide a connection between nonsymmetric networks and Cayley graphs. In particular, they showed that the shuffle-exchange and de Bruijn networks can be seen as group action graphs of cube-connected cycles and butterfly networks, respectively. This approach provides efficient emulation of many communication algorithms in these Cayley graphs through the smaller group action graphs associated with them. Similar ideas have been used in [7, 11] to study permutation networks. By using *arc-colorings* (or 1-factorizations), each regular digraph Γ is associated to a number of Cayley digraphs. The digraph Γ is a group action graph associated to each of these Cayley graphs, which we will call *Cayley regular covers* of Γ. The formal definitions are given in section 2.

*Received by the editors March 11, 1996; accepted for publication April 15, 1997. This research was supported in part by the Spanish Research Council CICYT under grant TIC 94-0592 and by the Catalan Research Council, CIRIT.

http://www.siam.org/journals/sidma/11-2/28864.html

†Departament de Matemàtica Aplicada i Telemàtica, Universitat Politècnica de Catalunya, Jordi Girona 1, Edifici C-3, 08034 Barcelona, Spain (matmed@mat.upc.es, oriol@mat.upc.es).

In this paper, we analyze the Cayley digraphs associated in this way to the de Bruijn digraphs. As it has been shown in [3], one of them corresponds to the butterfly network. One of the many ways of describing the de Bruijn digraphs is as an iterated line digraph of a complete graph [10]. Using this fact, it can be seen that any Cayley regular cover of a de Bruijn digraph is also an iterated line digraph. This result is proved in section 3. For a wide family of arc-colorings of the de Bruijn digraphs which can be described in an algebraic way, the corresponding Cayley regular covers turn out to be iterated line digraphs of simple well-known digraphs, as it is shown in section 4. Actually, they correspond to the digraphs studied by Praeger in [22]. This knowledge enables us to determine their diameter, to describe simple shortest-path, self-routing algorithms, and yields a simple way to study many of their properties, such as hamiltonicity, fault-tolerance, girth, and automorphism group. These results can be applied in particular to butterfly networks. In addition, several kinds of Cayley digraphs with simple routing algorithms and similar properties can be obtained in this way.

The knowledge of the routing in the Cayley covers mentioned above can also be used to generate specific sets of permutations by using de Bruijn networks. All the arc-colorings produce transitive permutation groups. In section 5 we show that one of the Cayley covers obtained in this way is doubly transitive. Unfortunately, not all Cayley regular covers of the de Bruijn digraphs can be easily handled. Nevertheless, we show that there is always an arc-coloring of the de Bruijn digraph whose Cayley regular cover is a Cayley digraph on the whole symmetric group. This fact provides a simple proof of the rearrangeability of the shuffle network. However, the determination of its diameter and routing algorithms seems to be a difficult task.

In order to keep the paper as self-contained as possible, we include some of the terminology and basic results in section 2. Graph theoretical terms used but not defined there can be found in [30]. For group theoretical terms see, for instance, [24, 28].

2. Basic terminology and preliminary results. Let G be a group of permutations on a finite set V. The product of permutations $\sigma, \tau \in G$ is written from right to left; that is, $(\sigma\tau)(x) = \sigma(\tau(x))$. The group of all permutations on V is the *symmetric* group S_n on n symbols, $n = |V|$, which has cardinality $n!$.

Let G be a group. An affine transformation of G is a bijective map of the form $f(x) = gh(x)$, where h is an automorphism of the group and g is a fixed element. When dealing with affine transformations of G, the notion of semidirect product arises naturally. If H is a group of automorphisms of G, the semidirect product $G \times_\iota H$ is the group of pairs $(g, h) \in G \times H$ where the product is defined as

$$(g, h)(g', h') = (gh(g'), hh').$$

The semidirect product $G \times_\iota H$ can be thought of as a group of permutations (or affinities) acting on G as $(g, h)(x) = gh(x)$. Then the product just defined corresponds to the composition of permutations. A slightly more general notion of semidirect product is obtained when we allow H to be any group and $\pi : H \to Aut(G)$ a group homomorphism. Then the product in the semidirect product $G \times_\pi H$ is defined as

$$(g, h)(g', h') = (g\pi(h)(g'), hh').$$

Let us briefly introduce some terminology of graph theory. Throughout the paper, we consider finite directed graphs (*digraphs* for short) $\Gamma = (V, E)$, where V is the set of *vertices* and $E \subset V \times V$ is the set of *arcs*. The existence of (x, x) arcs called *loops*

is allowed and there are no multiple arcs. Given a vertex x, we denote by $\Gamma^j(x)$ the set of vertices reachable from x by a path of length j (we omit the superscript when $j = 1$.) All digraphs are supposed to be d-regular and (*strongly*)*connected*, that is, both the indegree and outdegree of every vertex are d and there is a path from x to y for all $x, y \in V$. The *diameter* of Γ is the minimum k such that $\cup_{j \le k} \Gamma^j(x) = V$ for all $x \in V$. The length of the shortest cycle is the *girth* of Γ. The connectivity of the digraph is the cardinal of the smallest set of vertices whose deletion yields a nonconnected digraph and it is denoted by $\kappa(\Gamma)$. For a d-regular digraph, we clearly have $\kappa(\Gamma) \le d$. A permutation of the vertices of Γ which preserves (directed) adjacencies is an *automorphism* of the digraph. The digraph Γ is *vertex transitive* when, for each pair x, y of vertices, there is a digraph automorphism f such that $f(x) = y$.

Let G be a finite group and S a generating subset of G with cardinality d. The (*left*) *Cayley digraph* $Cay(G, S)$ has G as set of vertices, and (x, y) is an arc whenever $y = sx$ for some $s \in S$. The Cayley digraph $Cay(G, S)$ is a vertex-transitive, strongly connected d-regular digraph.

Each arc (x, y) of the Cayley digraph $Cay(G, S)$ can be labeled with the element $s \in S$ such that $y = sx$. This is the first example of an *arc-coloring*. Given a digraph Γ and a set $C = \{c_1, \ldots, c_d\}$ of colors, an *arc-coloring* of Γ with the set C is an assignment of a color in C to each arc of Γ in such a way that, for any vertex x, the arcs which are incident to x have different colors and the arcs which are incident from x have different colors as well. The couple (Γ, C) is said to be an *arc-colored* digraph.

It is well known that, as a consequence of Hall's matching theorem, every d-regular digraph admits an arc-coloring with d colors.

Given an arc-coloring of a d-regular digraph Γ with the set $C = \{c_1, \ldots, c_d\}$, the set of arcs of color c_i can be identified with the permutation of V, also denoted by c_i, such that $c_i(x) = y$ if and only if (x, y) is an arc with color c_i. In this way, C is a set of permutations of V such that $(x, c(x)) \in E$ for all $x \in V, c \in C$ and if $c(x) = c'(x)$ for any $x \in V$, then $c = c'$. We say that C is a *decomposition into permutations* of Γ. The *permutation group* of (Γ, C) is the subgroup $P(\Gamma, C)$ of $Sym(V)$ generated by C. In what follows we will refer to the set C as both a set of colors and a set of permutations. We also refer to *colorings* to mean *arc-colorings*.

The *line digraph* $L\Gamma$ of a d-regular digraph $\Gamma = (V, E)$ has E as vertex set, and $(x, y) \in E$ is adjacent to $(y', z) \in E$ if and only if $y = y'$. Hence, $L\Gamma$ is also d-regular and has order $d|V|$. We recursively define $L^k\Gamma = L(L^{k-1}\Gamma)$ for $k \ge 1$ and $L^0\Gamma = \Gamma$. Note that the vertices of $L^k\Gamma$ correspond to the paths of length k of Γ. In the next propositions we give the Heuchene's characterization of line digraphs [15] and some basic results related to them.

THEOREM 2.1 (see [15]). *A d-regular digraph $\Gamma = (V, E)$ is a k-iterated line digraph of another digraph Γ' if and only if the following conditions hold:* (i) *for every two vertices x, y and every $1 \le i \le k$, either $\Gamma^i(x) = \Gamma^i(y)$ or $\Gamma^i(x) \cap \Gamma^i(y) = \emptyset$;* (ii) *for every vertex x, $|\Gamma^k(x)| = d^k$.*

THEOREM 2.2 (see [4, 10]). *Let Γ be a connected d-regular digraph, $d \ge 2$. Then* (i) *$\kappa(L\Gamma) \ge \kappa(\Gamma)$;* (ii) *$diam(L\Gamma) = diam(\Gamma) + 1$;* (iii) *$L\Gamma$ is a Hamiltonian digraph;* (iv) *$Aut(L\Gamma) \simeq Aut(\Gamma)$;* (v) *$L\Gamma$ has the same girth as Γ.*

The above proposition is the basis for the use of the line digraph technique to provide infinite families of digraphs with nice properties concerning diameter and connectivity.

The existence of routings is also preserved by the line digraph operation. A *shortest-path, self-routing* in a digraph is a map $\rho : V \times V \to V$ such that, for $x \ne y$,

$\rho(x,y) \in \Gamma(x)$ and $d(\rho(x,y),y) = d(x,y) - 1$, (and $\rho(y,y) = y$). In other words, given a destination vertex y, such a routing provides a path of minimum length to y in such a way that each step of the path can be computed with the only knowledge of the current position and the destination vertex y. From the algorithmic point of view, the efficiency of such routings relies on the existence of a simple local procedure to perform this computation rather than having a space consuming table of ρ stored at each node. The self-routing schemes we consider are equipped with such a local procedure.

If ρ is a shortest-path, self-routing in a digraph Γ, the *induced* routing ρ_L in its line digraph $L\Gamma$ is the map $\rho_L : E \times E \to E$ defined as $\rho_L((x,y),(u,v)) = (y,\rho(y,u))$ for $y \neq u$, $(x,y) \neq (u,v)$, and $\rho_L((x,u),(u,v)) = (u,v)$ (and $\rho_L((u,v),(u,v)) = (u,v)$). The following result is straightforward.

PROPOSITION 2.3. *If ρ is a shortest-path, self-routing in a digraph Γ, then the induced routing ρ_L in its line digraph $L\Gamma$ is also a shortest-path, self-routing.*

The *de Bruijn digraphs*, denoted by $B(d,k)$, introduced in [6], have proved to be worthwhile because of their good behavior in many applications. The vertex set is the set of d^k words $x_0x_1,\ldots,x_{k-1}$ of length k, on an alphabet Δ of d symbols, and every word $x_0x_1,\ldots,x_{k-1}$ is adjacent to the d words $x_1,\ldots,x_{k-1}x_k$, with $x_k \in \Delta$. It has been shown in [10] that the de Bruijn digraphs are iterated line digraphs of the complete symmetric digraphs with loops

$$B(d,k) = L^{k-1}K_d^+.$$

Then, by Theorem 2.2, it easily follows that $B(d,k)$ has connectivity $d-1$, diameter k, is hamiltonian and $Aut\,(B(d,k)) \simeq S_d$. Moreover, the trivial shortest-path, self-routing in K_d^+ gives rise to a shortest-path, self-routing in each of the $B(d,k), k > 1$.

3. Regular Cayley covers of arc-colored digraphs. Let Γ be a d-regular connected digraph, $d \geq 2$, and let $C = \{c_1,\ldots,c_d\}$ be a coloring of Γ. Let $P = P(\Gamma,C)$ be the permutation group on the set of vertices of Γ generated by the permutations in C. The *Cayley regular cover* of (Γ,C) is the Cayley digraph $Cay(P,C)$.

As it has been pointed out in [3], Γ is a quotient digraph of each of its Cayley regular covers. Indeed, let P_x be the stabilizer in P of a vertex $x \in V$ (i.e., the set of permutations in P for which x is a fixed point). Then the map $f : Cay(P,C) \to \Gamma$, defined such that $f(\sigma) = \sigma(x)$ for every $\sigma \in P$, is a graph homomorphism onto Γ. Moreover, for every $y \in V$, $f^{-1}(y)$ has the cardinality of P_x and, if x is adjacent to y in Γ, then there is a perfect matching from $f^{-1}(x)$ to $f^{-1}(y)$ in $Cay(P,C)$. This fact is illustrated in Figure 1 which shows a coloring of the digraph $B(2,2)$ together with its Cayley regular cover.

We are interested in a particular property of Γ which stands in any of its Cayley regular covers.

THEOREM 3.1. *Let Γ be a k-iterated line digraph of a d-regular digraph Γ', and let C be an arc-coloring of Γ. Then the Cayley regular cover of (Γ,C) is also a k-iterated line digraph of a d-regular digraph.*

Proof. We use the characterization of Theorem 2.1. Let $X = Cay(P,C)$ be the Cayley regular cover of (Γ,C) and let σ,σ' be two permutations in P such that $\Gamma_X^i(\sigma) \cap \Gamma_X^i(\sigma') \neq \emptyset$, $i \leq k$. By symmetry, we can suppose that $\sigma' = \iota$, the identity permutation. Then $\sigma = \tau^{-1}\tau'$ for some $\tau,\tau' \in C^i$. For each vertex x of Γ, $\tau'(x) = \tau\sigma(x) \in \Gamma^i(x) \cap \Gamma^i(\sigma(x))$. Since Γ is a k-iterated line digraph, it follows that $C^i(x) = \Gamma^i(x) = \Gamma^i(\sigma(x)) = C^i(\sigma(x))$. As the choice of x is arbitrary, we get $\Gamma_X^i(\iota) = C^i = C^i\sigma = \Gamma_X^i(\sigma)$, which is condition (i) of Theorem 2.1 for X. On the other hand,

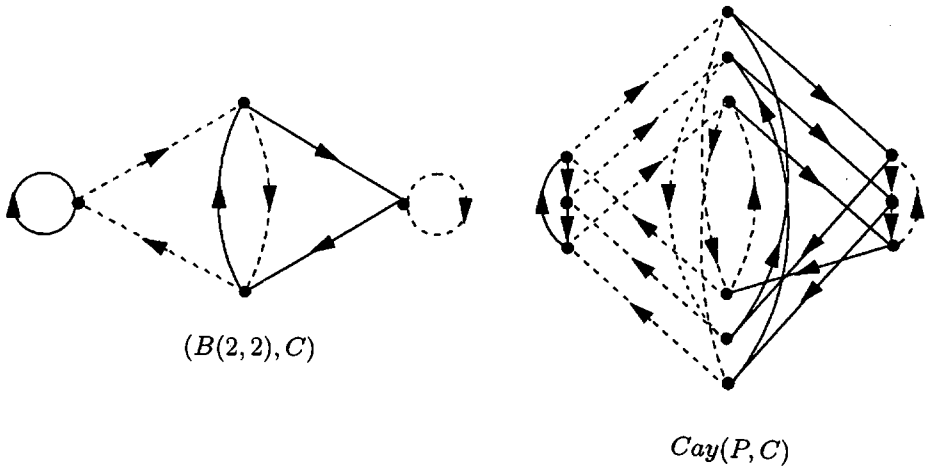


FIG. 1. *An arc-coloring of* $B(2,2)$ *and its Cayley regular cover.*

suppose that $|C^k| < d^k$. Then we would have two paths of length k starting at each vertex x of Γ with a common endpoint, which contradicts the fact that $|\Gamma^k(x)| = d^k$. Therefore, X is a k-iterated line digraph. □

According to the above theorem, for each coloring C of $\Gamma = L^k\Gamma'$, there exists a d-regular digraph X' such that $L^kX' = X$, where X is the Cayley regular cover of (Γ, C). We put $\Gamma' = L^{-k}\Gamma$ and $X' = L^{-k}X$. It is also worth noting that, since X is vertex transitive, X' has an automorphism group which is transitive on the paths of length k. In particular, X' is arc-transitive and therefore has maximum connectivity; see [13]. As a consequence of Theorem 2.2, we can state the following corollary.

COROLLARY 3.2. *Let* C *be a coloring of* $\Gamma = L^k\Gamma'$, *let* X *be the Cayley regular cover of* (Γ, C) *and let* $X' = L^{-k}X$. *Then* (i) X *has maximum connectivity;* (ii), *diam*(X)=*diam*$(X') + k$.

As a matter of fact, Theorem 2.2 provides much more information on a Cayley regular cover X through the knowledge of $X' = L^{-k}X$, such as the determination of its automorphism group and the existence of routings. In the next sections we use those facts for the study of certain regular Cayley covers of the de Bruijn digraphs.

4. Cayley covers of the de Bruijn digraphs. In this section, we study some particular colorings of the de Bruijn digraphs which can be described in an algebraic way. Let us identify the set of vertices of the de Bruijn digraph $B(d,k)$ with the elements of the group $(Z_d)^k = \underbrace{Z_d \times \cdots \times Z_d}_{k}$. The following lemma is straightforward.

LEMMA 4.1. *For any map* $\lambda : (Z_d)^{k-1} \to Z_d$, *the set* $C_\lambda = \{\lambda_0, \ldots, \lambda_{d-1}\}$ *of maps on* $(Z_d)^k$ *defined as*

$$\lambda_i(y, \mathbf{x}) = (\mathbf{x}, \lambda(\mathbf{x}) + y + i), \quad \mathbf{x} \in (Z_d)^{k-1}, \qquad i, y \in Z_d,$$

is an arc-coloring of $B(d,k)$.

We call *linear colorings* those obtained as in the above lemma when λ is a group homomorphism. Figure 2 shows the four linear colorings of $B(2,3)$.

It is worth noting that the two permutations in the linear coloring associated with the null map $\lambda \equiv 0$ are the *shuffle* and *shuffle-exchange* permutations on 2^k symbols (see [27]).

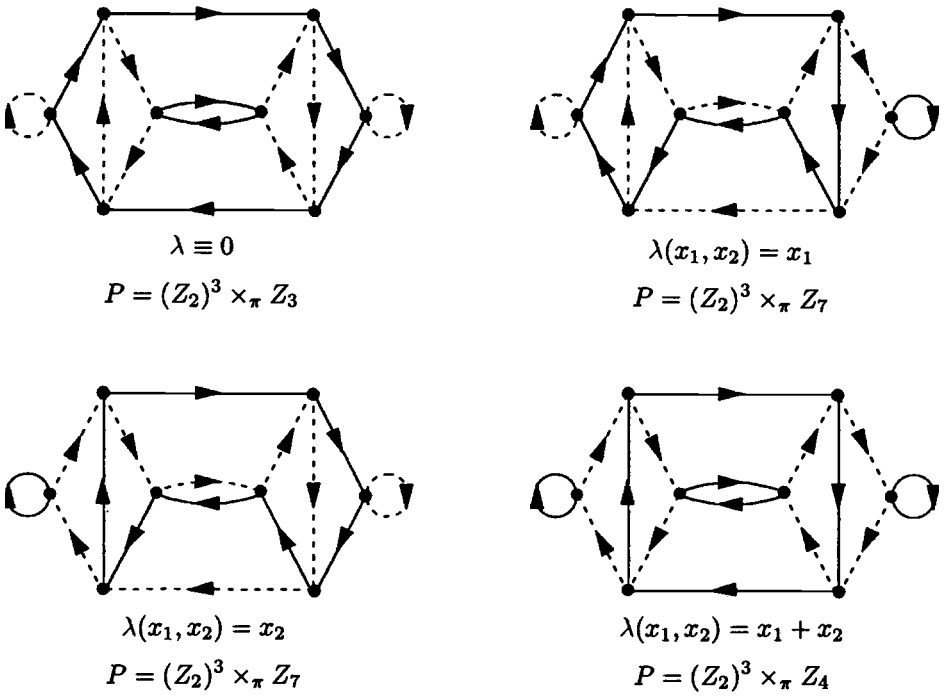


FIG. 2. *Linear colorings of* $B(2,3)$.

The permutation group generated by each of the linear colorings of the de Bruijn digraphs was computed in [26]. Notice that, for any linear coloring C_λ, the permutation λ_0 is actually a group automorphism of $(Z_d)^k$.

PROPOSITION 4.2 (see [26]). *The permutation group generated by the linear coloring* C_λ *of* $B(d,k)$ *is isomorphic to the semidirect product* $(Z_d)^k \times_\pi Z_m$, *where* m *is the order of the permutation* λ_0 *and* $\pi : Z_m \to Aut(Z_d)^k$ *is defined as* $\pi(i) = \lambda_0^i$.

The different groups obtained from the linear colorings of $B(2,3)$ are also depicted in Figure 2. It is easy to see each element of the group $(Z_d)^k \times_\pi Z_m$ as a permutation of the vertices of $B(d,k)$ if we define the action of the group on $(Z_d)^k$ as

$$(\mathbf{a}, i)(\mathbf{x}) = \mathbf{a} + \lambda_0^i(\mathbf{x}), \qquad \mathbf{x} \in (Z_d)^k, (\mathbf{a}, i) \in (Z_d)^k \times_\pi Z_m.$$

For positive integers d and k, the *butterfly digraph* $But(d,k)$ is defined as having the set $(Z_d)^k \times Z_k$ as vertices. Following the notation in [3, 19], each subset $(Z_d)^k \times \{i\}$ is referred to as the ith level of the digraph, and each vertex $v = (x_0, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{k-1}; i)$ of the ith level is adjacent to the d vertices $w_j = (x_0, \ldots, x_{i-1}, x_i + j, x_{i+1}, \ldots, x_{k-1}; i+1)$, $j = 1, \ldots, d$ in the level $i+1$. Figure 3 shows the digraph $But(2,3)$. By using the null map $\lambda \equiv 0$ we obtain the following characterization of the butterfly networks.

PROPOSITION 4.3. *For all positive integers* d *and* k, *the butterfly network* $But(d,k)$ *is isomorphic to the Cayley regular cover of the de Bruijn digraph* $B(d,k)$ *with the linear coloring* C_0.

Proof. When λ is the null map, λ_0 consists simply of a cyclic shift of the coordinates of every element in $(Z_d)^k$. Therefore, the order of λ_0 is k. Then, by Proposition 4.2,

$$P(B(d,k), C_0) \simeq (Z_d)^k \times_\pi Z_k.$$

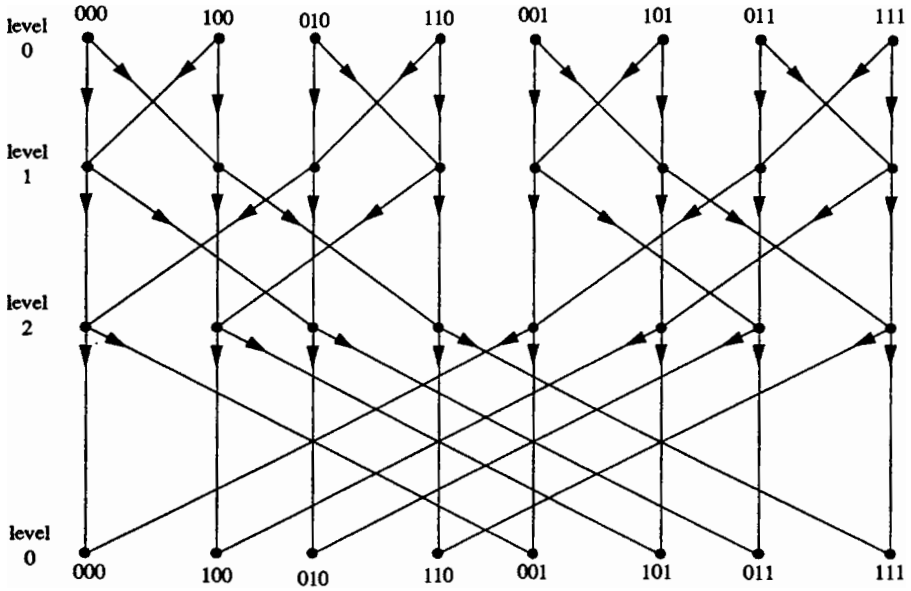


FIG. 3. *The butterfly digraph* $But(2,3)$.

Let us now identify the vertices of the Cayley regular cover $Cay(B(d,k),C_0)$, with the vertices of the butterfly $But(d,k)$, by performing a cyclic shift of i positions to the right on the coordinates of each node in the ith level:

$$(x_0,\ldots,x_{k-1};i) \leftrightarrow (x_{k-i},\ldots,x_{k-1},x_0,\ldots,x_{k-i-1};i).$$

Note that the permutation λ_j of the arc-coloring C_0 corresponds to the element $(0,\ldots,0,j;1)$ in $P(B(d,k),C_0)$ so that the vertex $(x_0,\ldots,x_{k-1};i)$ is adjacent to the vertices

$$(0,\ldots,0,j;1)(x_0,\ldots,x_{k-1};i) = (x_1,\ldots,x_{k-1},x_0+j;i+1)$$

in the Cayley regular cover. Therefore, the former identification is actually a digraph isomorphism. □

Figure 4 shows $But(2,3)$ as a Cayley cover of $B(2,3)$. In addition to the butterfly networks, the linear colorings provide a whole family of Cayley digraphs associated with the de Bruijn digraphs which have interesting structural properties. Recall that the de Bruijn digraph $B(d,k)$ can be seen as the $(k-1)$-iterated line digraph of the complete symmetric digraph with loops K_d^+. Furthermore, by Theorem 3.1 each of the Cayley regular covers of $B(d,k)$ is also a $(k-1)$-iterated line digraph. We next show that for Cayley regular covers $X_\lambda(d,k)$ associated to linear colorings, the digraph $L^{-(k-1)}X_\lambda(d,k)$ turns out to be a very simple digraph.

For positive integers d and m, the *complete generalized cycle* $C(d,m)$ is the Cayley digraph $Cay(Z_d \times Z_m, \{(0,1),\ldots,(d-1,1)\})$. The complete generalized cycle $C(2,k)$ is shown in Figure 5.

THEOREM 4.4. *Let C_λ be a linear coloring of the de Bruijn digraph $B(d,k)$. Then the Cayley regular cover of $(B(d,k),C_\lambda)$ is the $(k-1)$-iterated line digraph of the complete generalized cycle $C(d,m)$, where m is the order of the permutation λ_0.*

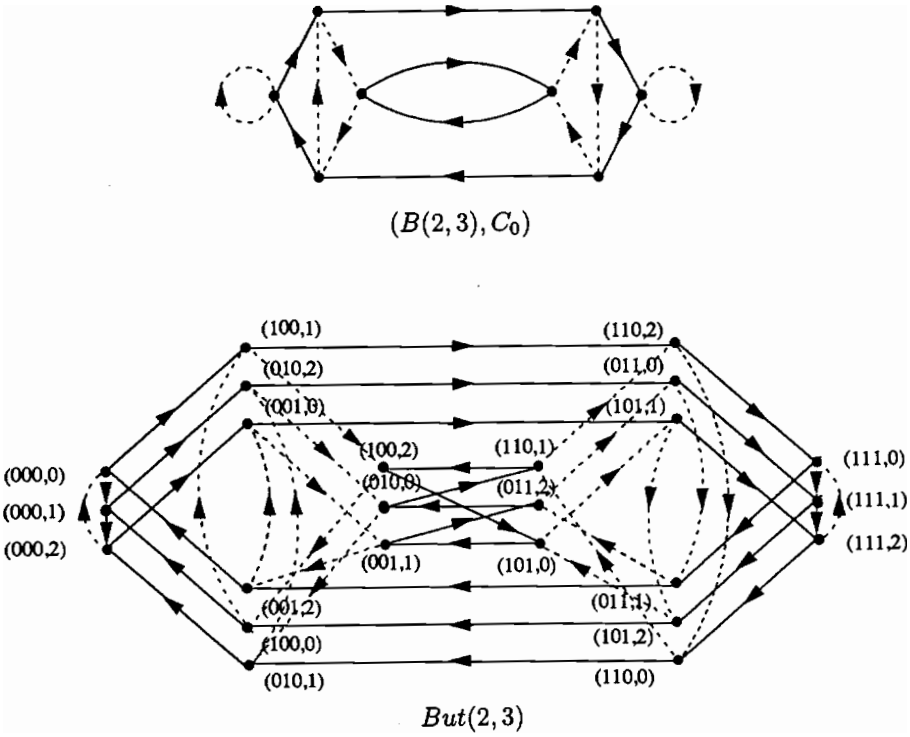


FIG. 4. *The arc-colored digraph* $(B(2,3),C_0)$ *and its Cayley regular cover* $But(2,3)$.

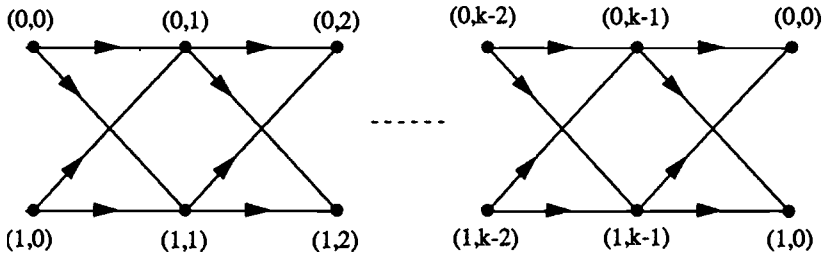


FIG. 5. *The complete generalized cycle* $C(2,k)$.

Proof. Let $C(d,m,k)$ denote the $(k-1)$-iterated line digraph of the complete generalized cycle $C(d,m)$. Each vertex of $C(d,m,k)$ corresponds to a path of length $k-1$ in $C(d,m)$. Let us identify the path $(x_0,i),(x_1,i+1),\ldots,(x_{k-1},i+k-1)$ with the element $(x_0,x_1,\ldots,x_{k-1};i)\in (Z_d)^k\times Z_m$. By Proposition 4.2, this identification provides a bijection between the vertices of the digraph $C(d,m,k)$ and the vertices of the Cayley regular cover of $(B(d,k),C_\lambda)$. It is straightforward to check that it is also a digraph isomorphism. Indeed, given $x=(x_0,\ldots,x_{k-1};i)$ the set $\Gamma(x)=\{(x_1,\ldots,x_{k-1},y;i+1),y\in Z_d\}$ in $C(d,m,k)$ coincides with

$$\Gamma(x)=\{(x_1,\ldots,x_{k-1},\lambda(x_1,\ldots,x_{k-1})+x_0+y;i+1),y\in Z_d\}$$

in $Cay((Z_d)^k\times_\pi Z_m,C_\lambda)$. □

As a consequence of the characterization in the above theorem, many of the properties of the Cayley regular covers of the de Bruijn digraphs can be easily deduced via Theorem 2.2 and Proposition 2.3.

THEOREM 4.5. *Let $X_\lambda(d,k)$ be the Cayley regular cover of the de Bruijn digraph $B(d,k)$ with the linear coloring C_λ, and let m be the order of the permutation $\lambda_0 \in C_\lambda$. Then* (i) *$X_\lambda(d,k)$ is maximally connected;* (ii) *$X_\lambda(d,k)$ is hamiltonian;* (iii) *$X_\lambda(d,k)$ has diameter $m+k-1$;* (iv) *the automorphism group of $X_\lambda(d,k)$ is isomorphic to $(S_d)^k \times_\pi Z_m$, where S_d is the symmetric group on d symbols and $\pi(i)$ is the cyclic shift of i positions on $(\sigma_1,\ldots,\sigma_d) \in (S_d)^k$;* (v) *$X_\lambda(d,k)$ has girth m; and* (vi) *$X_\lambda(d,k)$ has shortest-path self-routings.*

In fact, it is straightforward to check that the complete generalized cycle $C(d,m)$ is maximally connected and has diameter m. It was proved by Praeger in [22] that $C(d,m)$ has the automorphism group described in part (iv) of the above theorem. It is also easy to check that the map $\rho : (Z_d \times Z_m)^2 \to Z_d \times Z_m$, defined as $\rho((x,i),(y,j)) = (y,i+1)$ whenever $(x,i) \neq (y,j)$, is one of the many shortest-path routings which can be defined in $C(d,m)$, thus providing in a constructive way the shortest-path self-routing of $X_\lambda(d,k)$ via Proposition 2.3.

Actually, there is a broader collection of results concerning the line digraph than those described in section 2. We would like to mention at least one of them which is relevant to the applications in the context of networks. A maximally connected digraph is said to be *superconnected* when the only minimal disconnecting sets are the set of vertices adjacent to or adjacent from a vertex of the digraph. It has been proven in [9] that if a digraph is superconnected, then all its iterated line digraphs have the same property. On the other hand, the superconnected Cayley graphs on abelian groups were characterized in [14], from which it follows that the digraphs $C(d,m)$ are superconnected. Therefore, we have the following.

PROPOSITION 4.6. *For each λ, the Cayley regular cover $X_\lambda(d,k)$ is superconnected.*

Some of the results stated in Theorem 4.5 have been obtained in different ways for the case of the butterfly network; see for instance [3, 19]. One of the properties mentioned in [3] which can also be extended to the whole family of Cayley digraphs considered here is the existence of a disjoint family of complete d-ary trees, which follows from the fact that the de Bruijn digraph $B(d,k)$ contains a complete d-ary tree of depth $k-1$.

PROPOSITION 4.7. *For each λ, the Cayley regular cover $X_\lambda(d,k)$ contains m disjoint copies of the complete d-ary tree of depth $k-1$.*

5. Permutation networks. Not only are the Cayley regular covers interesting in themselves as network topologies with suitable properties, but they can also be seen as a model for different groups of permutations which can be performed without conflict in the original digraph. This approach is useful in the design of permutation networks. A permutation network is modeled as a digraph in which, at each time unit, the contents of every node can be transferred to one of its neighbors in such a way that no conflicts occur, that is, no two contents are sent to the same node. Therefore, at each unit time a permutation of the contents of the nodes is performed. Thus, the control of the network requires the knowledge of all permutations available and an algorithm to produce each of them in as few unit times as possible.

Colored digraphs provide a natural setting for the modeling of permutation networks. Given a colored digraph (Γ, C), every element σ of the permutation group $P(\Gamma, C)$ corresponds to a permutation of the vertices of Γ which can be performed as

a sequence of permutations in C, thus using the arcs of the digraph with no conflicts. In addition, a shortest-path routing in the Cayley regular cover $Cay(\Gamma, C)$ provides an algorithm to generate every such permutation, and its diameter gives a bound on the worst case time to generate them.

In this section we consider two examples of Cayley regular covers of the de Bruijn digraphs which are useful in this context. For simplicity we restrict ourselves to the case of degree two.

A permutation group P acting on a set V of n elements is said to be *k-transitive* if, for any pair $(x_1, \ldots, x_k), (y_1, \ldots, y_k) \in V^k$ of k-tuples of (different) elements of V, there is a permutation $\sigma \in P$ such that $\sigma(x_i) = y_i, i = 1, \ldots, k$. If such a permutation is unique for each pair of k-tuples, P is said to be *sharply* k-transitive. A k-transitive group P is sharply k-transitive if and only if it has $n!/(n-k)!$ elements. The symmetric group is the only (sharply) n-transitive group, and 2-transitive groups are usually referred to as *doubly* transitive groups.

The first example we consider consists of a linear coloring of the de Bruijn digraph $B(2, k)$ whose permutation group is sharply 2-transitive.

THEOREM 5.1. *Let $f(x) = x^k - a_{k-1}x^{k-1} - \cdots - a_1 x - a_0$ be a primitive polynomial of the Galois field $GF(2^k)$. Let the map $\lambda : (Z_2)^{k-1} \to Z_2$ be defined as $\lambda(x_1, \ldots, x_{k-1}) = a_{k-1}x_{k-1} + \cdots + a_1 x_1$. Then the permutation group of the de Bruijn digraph $B(2, k)$, $k \geq 2$, with the coloring C_λ, is sharply 2-transitive.*

Proof. The order of the permutation λ_0 of $(Z_2)^k$ defined as

$$\lambda_0(x_0, \mathbf{x}) = (\mathbf{x}, x_0 + \lambda(\mathbf{x})), \qquad \mathbf{x} \in (Z_2)^{k-1}$$

is the minimum common period of the recurrence whose characteristic polynomial is f. Therefore λ_0 has order $2^k - 1$ (see [21] for details). Hence, the 1-factor of $B(d, k)$ corresponding to λ_0 consists of a loop on the vertex $\mathbf{0} = (0, \ldots, 0)$ and a hypohamiltonian cycle. Therefore, by Proposition 4.2, the permutation group of C_λ is $P = P(B(2, k), C_\lambda) = (Z_2)^k \times_\pi Z_{2^k-1}$.

Let us denote by $\mathbf{0} = (0, \ldots, 0)$ and $\mathbf{1} = (1, \ldots, 1) \in (Z_2)^k$ the vertices of $B(2, k)$ in which the two loops of $B(2, k)$ are located. Let ϕ be the only nontrivial automorphism of the digraph $B(2, k)$, namely, $\phi(x_0, \ldots, x_{k-1}) = (x_0 + 1, \ldots, x_{k-1} + 1)$, where the addition is modulo 2. Let us show that $\phi\lambda_0\phi = \lambda_1$. Since f is a primitive polynomial, then $\sum_{i=0}^{k-1} a_i = 0 \bmod 2$ (otherwise f would have 1 as a root) and $a_0 = 1$. Therefore, a simple calculation gives

$$\phi\lambda_0\phi(x_0, \ldots, x_{k-1}) = \left(x_1, \ldots, x_{k-1}, x_0 - \sum_{i=1}^{k-1} a_i x_i - \sum_{i=1}^{k-1} a_i \right) = \lambda_1(x_0, \ldots, x_{k-1}).$$

As a consequence, the permutation λ_1 has the same cyclic structure as λ_0, so it contains the loop in $\mathbf{1}$ and a cycle of length $2^k - 1$.

Let $\mathbf{x}, \mathbf{y}$ be any pair of vertices of $B(2, k)$. For the 2-transitivity of the permutation group P, it suffices to show that there exists a permutation which sends $\mathbf{x}$ to $\mathbf{0}$ and $\mathbf{y}$ to $\mathbf{1}$. Let r be the length of the path from $\mathbf{x}$ to $\mathbf{0}$ through the long cycle of λ_1 (put $r = 0$ if $\mathbf{x} = \mathbf{0}$), and let s be the length of the path from $\lambda_1^r(\mathbf{y})$ to $\mathbf{1}$ through the long cycle of λ_0 (or $s = 0$ if $\lambda_1^r(\mathbf{y}) = \mathbf{1}$). Then the permutation $\sigma = \lambda_0^s \lambda_1^r$ sends $\mathbf{x}$ to $\mathbf{0}$ and $\mathbf{y}$ to $\mathbf{1}$. Moreover, since P has order $2^k(2^k - 1)$, the group is sharply 2-transitive. □

In the above proof, a factorization of length at most $2^k - 2$ is obtained for every element in the permutation group. Actually, by Theorem 4.5, the corresponding

Cayley regular cover has diameter $2^{k-1}+k-1$. In addition, the simple shortest-path, self-routing algorithm described in the paragraph below Theorem 4.5 can be used to explicitly obtain a factorization of minimum length of any given permutation in the group.

The problem of characterizing those permutation networks in which every permutation between its nodes can be performed has lead to some attempts to find a simple and unifying technique to establish the rearrangeability of a permutation network [18, 20, 29, 31]. If the permutation network is modeled by a colored digraph, this purpose is then translated into the problem of determining whether the associated permutation group is the whole symmetric group. In this context, we say that a coloring of a digraph is a *complete* coloring when the associated permutation group is the symmetric group of degree the order of the digraph. For instance, it is straightforward to check that the complete digraph admits a complete coloring. We next show that the de Bruijn digraphs of degree two admit complete colorings, with the only exception being $B(2,2)$.

THEOREM 5.2. *The de Bruijn digraph $B(2,k)$, $k>2$ admits a complete coloring C so that $P(B(2,k),C)=S_{2^k}$.*

Proof. Let us consider the digraph $B(2,k-1)$ with the coloring $C_\lambda=\{\lambda_0,\lambda_1\}$, where λ is the map in Theorem 5.1. Let us identify the vertices of $B(2,k)$ with the couples (x,λ_i), where x is a vertex of $B(2,k-1)$ and $\lambda_i\in C_\lambda$. Let $C'=\{c'_0,c'_1\}$ be the coloring of $B(2,k)$ defined as $c'_0(x,\lambda_i)=(\lambda_i(x),\lambda_i)$ and $c'_1(x,\lambda_i)=(\lambda_i(x),\lambda_{i+1})$, where the subscripts are taken modulo 2. In other words, two consecutive arcs with the same color in $(B(2,k-1),C_\lambda)$ induce an arc of color c'_0 in $B(2,k)$ (it is said that C' is the coloring uniformly induced by C_λ). Hence, the permutation c'_0 consists of two cycles of length $2^{k-1}-1$ and two loops.

We next show that c'_1 determines one cycle of length 2^k-2 and a swap, which corresponds to the digon of $B(2,k)$. Let us consider the cycle h of c'_1 containing the vertex $(0,\lambda_0)$. By the definition of c'_1, h must have even length, say $2r$. Then

$$(c'_1)^{2r}(0,\lambda_0)=((\lambda_1\lambda_0)^r(0),\lambda_0)=((\lambda_1\phi)^{2r}(0),\lambda_0)=(((\lambda_1\phi)^2)^r(0),\lambda_0),$$

where, as in the proof of Theorem 5.1, ϕ is the nontrivial automorphism of $B(2,k)$ and $\phi\lambda_0\phi=\lambda_1$. As shown in the proof of Theorem 5.1, λ_1, and hence $\lambda_1\phi$, consists of a cycle of (odd) length $2^{k-1}-1$ and a fixed point. Thus $(\lambda_1\phi)^2$ also has this cyclic structure. Therefore r must be $2^{k-1}-1$ and the cycle h has length 2^k-2. The remaining cycles of c'_1 must then be either a swap or two loops. Since h contains the vertex $(0,\lambda_0)$, in which there is a loop of $B(2,k)$, our claim is proved.

Let us consider the coloring $C=\{c_0,c_1\}$ obtained by exchanging the colors in C' of the set of arcs which are incident to and from the vertex $\mathbf{0}$ and the arc $((1,0,\ldots,0),(0,\ldots,0,1))$. Thus, c_0 contains a cycle of length 2^{k-1}, a cycle of length $2^{k-1}-1$, and a loop in vertex $\mathbf{1}$, and c_1 contains a cycle of length 2^k-3, a swap, and a loop in vertex $\mathbf{0}$ (see Figure 6.)

We next show that the permutation group $P=P(B(2,k),C)$ is doubly transitive. Let us denote by $\mathbf{d_0}=(0101,\ldots)$ and $\mathbf{d_1}=(1010,\ldots)$ the vertices of the digon of $B(2,k)$. Let $\mathbf{x},\mathbf{y}$ be any pair of vertices in $B(2,k)$. Then there exists a permutation σ in P such that $\sigma(\mathbf{x})=\mathbf{d_0}$ and $\sigma(\mathbf{y})=\mathbf{d_1}$. Indeed, let c_0^1 denote the cycle of c_0 of length $2^{k-1}-1$ and let c_0^0 denote the cycle of length 2^{k-1}. First, let us suppose that $\mathbf{x},\mathbf{y}$ belong to different cycles in c_0 and $\mathbf{x},\mathbf{y}\neq\mathbf{1}$. Then there exists an integer r such that $c_0^r(\mathbf{x})=\mathbf{d_i}$ and $c_0^r(\mathbf{y})=\mathbf{d_{i+1}}$, since the cycles in c_0 have relatively prime lengths and each of them contains one vertex of the digon. We may also need to apply c_1 to get the proper position of the vertices $\mathbf{x},\mathbf{y}$ in the digon.

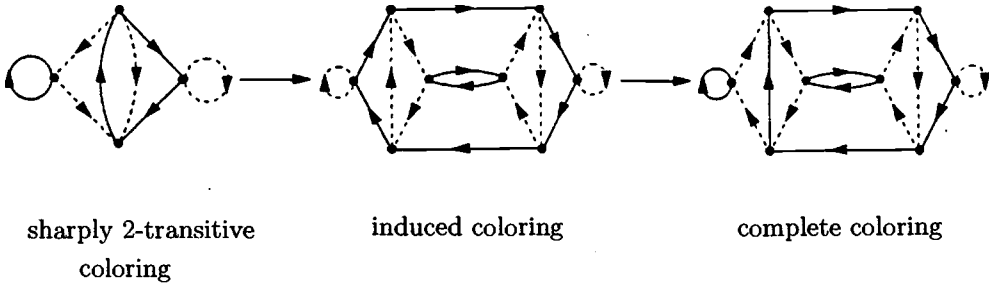


FIG. 6. *The construction of the complete coloring of* $B(2,3)$.

Secondly, if both vertices $\mathbf{x}, \mathbf{y}$ belong to the cycle c_0^0, then there exists an integer r such that $c_0^r(\mathbf{x}) = \mathbf{0}$. By applying the permutation c_1, which fixes $\mathbf{x}$, since the digraph is strongly connected the vertex $\mathbf{y}$ can be placed in the cycle c_0^1 and the former situation applies. Finally, in the remaining cases, by applying successively the permutation c_1 the vertices $\mathbf{x}, \mathbf{y}$ are located in one of the preceding situations.

As it is well known, a doubly transitive permutation group which contains a transposition is the symmetric group; see for instance [28, Thms. 9.6 and 13.3]. Therefore, since P is doubly transitive and contains the swap $c_1^{2^k-3} = (\mathbf{d_0}, \mathbf{d_1})$, we get $P = S_{2^k}$. □

Clearly, the proof of the above theorem is not valid for $B(2,2)$. Actually, it can be easily checked that the only two colorings of $B(2,2)$ are both linear so that none of them is complete.

Unfortunately, the problem of determining the diameter and routings of the Cayley cover of the complete coloring of $B(2,k)$ does not seem to be easy to handle. For $k = 3$, there is only one complete coloring which is shown in Figure 6 and corresponds to the one described in the above proof. The diameter of the corresponding Cayley cover, computed experimentally, turns out to be 22.

As a final remark, we would like to mention that the techniques used in this paper to obtain and analyze Cayley digraphs can also be applied to other families of iterated line digraphs. In addition to the de Bruijn digraphs, the iterated line digraphs of the complete symmetric digraphs without loops, also known as *Kautz* digraphs [16], have often been considered as models for networks. In many respects, Kautz digraphs enjoy even better properties regarding the diameter or fault-tolerance than their companions, the de Bruijn digraphs. Again, the use of Cayley regular covers may allow us to overcome their main drawback, namely, lack of symmetry. The study of some of the Cayley regular covers of Kautz digraphs is the object of a forthcoming paper by the authors.

Acknowledgments. The authors would like to thank the referees for their valuable remarks.

REFERENCES

[1] S. B. AKERS AND B. KRISHNAMURTHY, *Group graphs as interconnection networks*, in Proc. 14th Internat. Conf. on Fault-Tolerant Computing, 1984, pp. 422–427.

[2] S. B. AKERS AND B. KRISHNAMURTHY, *A group theoretic model for symmetric interconnection networks*, IEEE Trans. Comput., 38 (1989), pp. 555–565.

[3] F. ANNEXSTEIN, M. BAUMSLAG, AND A. L. ROSENBERG, *Group action graphs and parallel architectures*, SIAM J. Comput., 19 (1990), pp. 544–569.

[4] L. W. Beinecke and R. J. Wilson, *Selected Topics in Graph Theory* I, Academic Press, London, 1978.
[5] J. C. Bermond and C. Peyrat, *De Bruijn and Kautz networks: A competitor for the hypercube?*, Hypercube Distrib. Comput., 3 (1989), pp. 279–293.
[6] N. G. de Bruijn, *A combinatorial problem*, Konink. Nederl. Akad. Wetersh. Verh. Afd. Natuurk. Eerste Reelss, A49 (1946), pp. 758–764.
[7] M. Espona, *Xarxes de permutacions i digrafs acolorits: anàlisi i disseny*, Ph.D. thesis, Departament de Matemàtica Aplicada i Telemàtica, Universitat Politècnica de Catalunya, Barcelona, Spain, 1994 (in Catalan).
[8] V. Faber, *Cycle prefix digraphs for symmetric interconnection networks*, Networks, 23 (1993), pp. 641–649.
[9] M. A. Fiol and A. S. Lladó, *The partial line digraph technique in the design of large interconnection networks*, IEEE Trans. Comput., 41 (1992), pp. 848–857.
[10] M. A. Fiol, J. L. A. Yebra, and I. Alegre, *Line digraph iterations and the (d, k) digraph problem*, IEEE Trans. Comput., 33 (1984), pp. 400–403.
[11] M. A. Fiol, J. Fàbrega, O. Serra, and J. L. A. Yebra, *A unified approach to the design and control of dynamic memory networks*, Parallel Process. Lett., 3 (1993), pp. 445–456.
[12] E. Fuller and B. Krishnamurthy, *Symmetries in Graphs: An Annotated Bibliography*, Technical Report CR-86-03, Computer Research Laboratory, Tektronics Laboratories, Beaverton, OR, 1986.
[13] Y. O. Hamidoune, *Sur les atomes d'un graphe orienté*, C.R. Acad. Sci. Paris A, 1977, pp. 1253–1256.
[14] Y. O. Hamidoune, A. S. Llado, and O. Serra, *Vosperian and superconnected abelian Cayley digraphs*, Graphs Combin., 7 (1991), pp. 143–152.
[15] C. Heuchene, *Sur une certaine correspondance entre graphes*, Bull. Soc. Roy. Sci. Liège, 33 (1964), pp. 743–753.
[16] W. H. Kautz, *Design of optimal interconnection networks for multiprocessors*, in Architecture and Design of Digital Computers, NATO Advanced Summer Institute, 1969, pp. 249–272.
[17] S. Lakshmivarahan, J.-S. Jwo, and S. K. Dhall, *Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey*, Parallel Comput., 19 (1993), pp. 361–407.
[18] D. H. Lawrie, *Access and alignment of data in an array processor*, IEEE Trans. Comput., 24 (1975), pp. 1145–1155.
[19] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan-Kaufmann, San Francisco, CA, 1992.
[20] J. Lenfant, *Parallel permutations of data: A Benes network control algorithm for frequently used permutations*, IEEE Trans. Comput., 27 (1978), pp. 633–647.
[21] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, Cambridge University Press, Cambridge, 1986.
[22] C. E. Praeger, *Highly arc transitive digraphs*, European. J. Combin., 10 (1989), pp. 281–292.
[23] F. P. Preparata and J. E. Vuillemin, *The cube-connected cycles: A versatile graph for parallel computation*, J. ACM, 24 (1981), pp. 300–309.
[24] D. J. S. Robinson, *A Course in the Theory of Groups*, Graduate Texts in Mathematics, 80, Springer-Verlag, New York, 1982.
[25] Y. Saad and M. H. Schultz, *Topological properties of hypercubes*, IEEE Trans. Comput., 37 (1988), pp. 867–872.
[26] O. Serra and M. A. Fiol, *Arc-colored line digraphs and their groups*, in Graph Theory, Combinatorics, Algorithms, and Applications, Proc. Appl. Math. 54, Y. Alavi et al., eds., SIAM, Philadelphia, PA, 1991, pp. 459–468.
[27] H. S. Stone, *Parallel processing with the Perfect Shuffle*, IEEE Trans. Comput., 20 (1971), pp. 153–161.
[28] H. Wielandt, *Finite Permutation Groups*, Academic Press, New York, 1964.
[29] C. L. Wu and T. Feng, *The universality of the shuffle-exchange network*, IEEE Trans. Comput., 30 (1981), pp. 324–332.
[30] H. P. Yap, *Some Topics in Graph Theory*, London Math. Soc. Lecture Notes Ser. 108, Cambridge University Press, Cambridge, 1986.
[31] Y.-M. Yeh and T. Feng, *On a class of rearrangeable networks*, IEEE Trans. Comput., 41 (1992), pp. 1361–1379.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 318–329, May 1998

011

A GREEDY ALGORITHM ESTIMATING THE HEIGHT OF RANDOM TREES*

TOMASZ ŁUCZAK†

Abstract. The behavior of a greedy algorithm which estimates the height of a random, labelled rooted tree is studied. A self-similarity argument is used to characterize the limit distribution of the length H of the path found by such an algorithm in a random rooted tree as the unique solution of an integral equation. Furthermore, it is shown that

$$\lim_{n\to\infty} \frac{\mathrm{E}H}{\sqrt{n}} = \frac{\sqrt{2\pi}}{2\sqrt{2} - \ln(3+2\sqrt{2})} = 2.352139\ldots ,$$

i.e., the expected length of the path constructed by the algorithm is roughly 93.8% of the expected height of a random rooted tree.

Key words. random trees, greedy algorithms

AMS subject classifications. 68Q25, 05C80, 05C05

PII. S0895480193258960

1. Introduction.

Let T_n be a random, labelled rooted tree on the vertex set $[n] = \{1, 2, \ldots, n\}$ with the root $v_0 \in [n]$. (Here and below we assume for convenience that the root is always the vertex number 1.) The limit distribution of the height $\tilde{H} = \tilde{H}(n)$ of T_n was determined by Rényi and Szekeres [5] who proved the following result.

THEOREM 0. *For every constant $\beta > 0$,*

$$\lim_{n\to\infty} (\tilde{H} = \lfloor \sqrt{2n}/\beta \rfloor) = 2\sqrt{\frac{2\pi}{n}}\beta^2 \sum_{i=1}^{\infty} \left(2i^4\pi^4\beta - 3i^2\pi^2\right) \exp(-\beta\pi^2 i^2) \tag{1}$$

$$= \sqrt{\frac{8}{n\beta}} \sum_{i=1}^{\infty} \left(\frac{2i^4}{\beta} - 3i^2\right) \exp\left(-\frac{i^2}{\beta}\right) ,$$

where the convergence is uniform for $\beta \in (c, C)$, for every constant $C > c > 0$.

Furthermore, they proved that the sth moment of $\tilde{H}/\sqrt{2n}$ tends to $2\Gamma(s/2+1)(s-1)\sum_{i=1}^{\infty} i^{-s}$. In particular, for the expectation and the variance of $\tilde{H}$, they got

$$\lim_{n\to\infty} \frac{\mathrm{E}\,\tilde{H}}{\sqrt{n}} = \sqrt{2\pi} = 2.50663\ldots \tag{2}$$

and

$$\lim_{n\to\infty} \frac{\mathrm{Var}\,\tilde{H}}{n} = \frac{2\pi(\pi-3)}{3} = 0.29655\ldots .$$

(see Flajolet et al. [2] for the generalization of these results to other simply generated families of trees).

*Received by the editors November 23, 1993; accepted for publication April 15, 1997. This research was partially supported by KBN grant 2 1087 91 01.
http://www.siam.org/journals/sidma/11-2/25896.html
†Mathematical Institute of the Polish Academy of Sciences, Poznań, Poland (tomasz@math.amu.edu.pl). Permanent address: Department of Discrete Mathematics, Adam Mickiewicz University, ul. Matejki 48/49, 60-769 Poznań, Poland.

For a tree T with the root v_0, let $\mathcal{F}(T)$ be the forest of rooted trees obtained from T by removing v_0 where, as the root of a tree $T' \in \mathcal{F}(T)$ we take the vertex adjacent to v_0 in T. The height of a tree T can be estimated using the following simple algorithm GREEDY, which constructs a long path starting at the root. In the first step, the algorithm removes the root v_0 of $T^{(0)} = T$, chooses the largest tree $T^{(1)}$ from $\mathcal{F}(T^{(0)})$ (if there are more than one of them it picks one with the lexicographically first root), and appends its root to the path. This procedure is repeated until for some h the tree $T^{(h)}$ consists of a single vertex.

The purpose of this note is to apply a simple self-similarity idea to study the length $H = H(n)$ of the path found in a random tree by the above procedure. We characterize the limit distribution of H as the solution of some integral equation and show that the expected value of $H/\sqrt{n}$ tends to a constant a, where

$$a = \frac{\sqrt{2\pi}}{2\sqrt{2} - \ln(3 + 2\sqrt{2})} = 2.353139\ldots .$$

Thus, in average, GREEDY generates a path whose length is roughly 93.8% of the expected height of the tree. We should remark that similar results can be derived from Aldous' "continuum" approach to random trees; we comment on it further in the last section.

The structure of the note is the following. In the next section, we prove some purely combinatorial facts on so-called (k, l, m)-decompositions. Based on these results, we characterize the limit distribution of H as the solution of some integral equation (see section 3) and find the asymptotic value of the expectation of H (see section 4). We conclude with a few comments on the self-similarity method we applied.

2. (k, l, m)-decompositions. A (k, l, m)-*decomposition* (P, F, S, R) of $[n] = \{1, 2, \ldots, n\}$ is a quadruple of graphs, such that

1. P is a path $v_0 v_1, \ldots, v_{k-1}$ starting at $v_0 = 1$;
2. F is a forest of k trees on $n - l - m$ vertices such that vertices $v_0, v_1, \ldots, v_{k-1}$ belong to different trees;
3. S is a rooted tree with l vertices and root v_k;
4. R is a tree on m vertices rooted at v_{k+1}; and
5. $[n] = V(F) \cup V(S) \cup V(R)$ is a partition of $[n]$.

We say that a (k, l, m)-decomposition (P, F, S, R) is *contained* in T if T is the tree rooted at v_0 with vertices $1, 2, \ldots, n$ and edges $E(T) = E(P) \cup E(F) \cup E(S) \cup E(R) \cup \{\{v_{k-1}, v_k\}, \{v_k, v_{k+1}\}\}$. A (k, l, m)-decomposition (P, F, S, R) is a *subdecomposition* of a (k', l', m')-decomposition if both (P, F, S, R) and (P', F', S', R') are contained in the same tree and $P \subseteq P'$. We call a (k, l, m)-decomposition (P, F, S, R) *bad* if $\mathcal{F}(S)$ contains a tree with more than m vertices (or a tree with precisely m vertices with the label of the root smaller than the label of the root of R) and *good* otherwise. Finally, we say that a decomposition is *proper* if all its subdecompositions are good.

It is easy to see that the notion of decomposition emerges naturally in the analysis of the algorithm GREEDY described above. Indeed, suppose that the algorithm finds a path $P_h = v_0 v_1, \ldots, v_h$ in a tree $T = T^{(0)}$. Then, for every $k < h$, deleting from T edges $\{v_{k-1}, v_k\}$ $\{v_k, v_{k+1}\}$ splits the tree into three parts, F_k, S_k, and R_k, which, together with the path $P_k = v_0 v_1, \ldots, v_{k-1}$, form a proper decomposition contained in T. On the other hand, if a (k, l, m)-proper decomposition (P, F, S, R) is contained in T, the algorithm finds the path P in T by the $(k-1)$th step. Thus, $|T^{(k)}| = m$ if and only if $T = T^{(0)}$ contains some proper (k, l, m)-decomposition, and, since for a given k (or a given m) every tree contains at most one proper (k, l, m)-decomposition, to

study the behavior of the algorithm on random rooted trees one needs to estimate the probability that a random tree contains some proper (k,l,m)-decomposition of $[n]$.

Let $a(n;k,l,m)$ be the number of all (k,l,m)-decompositions of $[n]$, and let $\hat{a}(n;k,l,m)$ denote the probability that a random rooted tree contains a (k,l,m)-decomposition, i.e., $\hat{a}(n;k,l,m) = a(n;k,l,m)/n^{n-2}$.

FACT 1. *For $k \geq 1$ we have*

$$\begin{aligned}\hat{a}(n;k,l,m) &= \frac{n!}{n^{n-1}}\frac{l^{l-1}}{l!}\frac{m^{m-1}}{m!}k\frac{(n-l-m)^{n-l-m-k-1}}{(n-m-l-k)!}\\ &= \frac{1}{2\pi}\frac{n^{3/2}}{l^{3/2}m^{3/2}}\frac{k}{(n-m-l)^{3/2}}\exp\left(-\frac{k^2}{2(n-m-l)}\right)\\ &\quad\times\exp\left(O\left(\frac{k^3}{(n-m-l)^2}+\frac{k}{n-m-l}+\frac{1}{k}+\frac{1}{l}+\frac{1}{m}\right)\right),\end{aligned}$$

whereas

$$\hat{a}(n;0,l,m) = \begin{cases}\dfrac{n!}{n^{n-1}}\dfrac{l^{l-1}}{l!}\dfrac{(n-l)^{n-l-1}}{(n-l)!} & \text{if} \quad l+m=n,\\ 0 & \text{if} \quad l+m\neq n.\end{cases}$$

Proof. To build a (k,l,m)-decomposition we must divide the set $\{2,3,\ldots,n\}$ into four parts of $k-1$, $n-m-l-k$, l, and m elements, respectively, arrange the vertices of the first set in a path P in one of $(k-1)!$ ways, construct a rooted forest on the first two sets and vertex $v_0 = 1$ in one of $k(n-l-m)^{n-l-m-k-1}$ ways, and build rooted trees on each from the remaining two sets. Thus, using Stirling's formula, we get

$$\begin{aligned}\hat{a}(n;k,l,m) &= \frac{(n-1)!}{(k-1)!\,(n-m-l-k)!\,l!\,m!}(k-1)!\\ &\quad\times k(n-l-m)^{n-l-m-k-1}\frac{l^{l-1}m^{m-1}}{n^{n-2}}\\ &= \frac{n!}{n^{n-1}}\frac{l^{l-1}}{l!}\frac{m^{m-1}}{m!}k\frac{(n-l-m)^{n-l-m-k-1}}{(n-m-l-k)!}\\ &= \frac{1}{2\pi}\frac{kn^{3/2}}{l^{3/2}m^{3/2}}\frac{(n-l-m)^{n-l-m-k-1}}{(n-m-l-k)^{n-m-l-k+1/2}}\\ &\quad\times\exp(k-l+O(1/(n-m-l)+1/k+1/l+1/m))\\ &= \frac{1}{2\pi}\frac{n^{3/2}}{l^{3/2}m^{3/2}}\frac{k}{(n-m-l)^{3/2}}\exp\left(-\frac{k^2}{2(n-m-l)}\right)\\ &\quad\times\exp\left(O\left(\frac{k^3}{(n-m-l)^2}+\frac{k}{n-m-l}+\frac{1}{k}+\frac{1}{l}+\frac{1}{m}\right)\right).\end{aligned}$$

Similarly, for $l+m=n$ we have

$$\hat{a}(n;0,l,m) = \binom{n-1}{l-1}\frac{l^{l-2}(n-l)^{n-l-1}}{n^{n-2}} = \frac{n!}{n^{n-1}}\frac{l^{l-1}}{l!}\frac{(n-l)^{n-l-1}}{(n-l)!}. \qquad \square$$

For a given k and m, where $m < n/2$, we set

$$p_n(k,m) = \mathrm{Prob}(|T_n^{(k)}| \geq n/2 \wedge |T_n^{(k+1)}| = m). \tag{3}$$

Thus, $p_n(k,m)$ tells us about the joint distribution of k and $|T_n^{(k+1)}|$, when the size of $T_n^{(k+1)}$ first drops under $n/2$. The limit value of $p_n(k,m)$ is given by the following result, crucial for our further considerations.

LEMMA 2. *Let f be the function defined as*

$$f(x,y) = \frac{1}{2\pi}\int_{1/2-y}^{y} \frac{x}{t^{3/2}y^{3/2}(1-t-y)^{3/2}} \exp\left(-\frac{x^2}{2(1-y-t)}\right) dt, \tag{4}$$

where $x>0$ and $y\in(1/4,1/2)$. Then, for $y\in(1/4,1/2)$,

$$p_n(\lfloor x\sqrt{n}\rfloor, \lfloor yn\rfloor) = \frac{1+o(1)}{n^{3/2}} f(x,y)\ ,$$

where for every $\epsilon>0$ the quantity $o(1)$ tends to 0 uniformly for $x>\epsilon$ and $1/4+\epsilon< y<1/2-\epsilon$.

Furthermore, there exists a constant C such that for every $x>0$ and $m\le n/4+\sqrt{n}$ we have

$$p_n(\lfloor x\sqrt{n}\rfloor, m) \le Cn^{-5/3}.$$

Proof. Let k denote the minimum value of k such that $|T_n^{(k+1)}|\le n/2$, and let $m=|T_n^{(k+1)}|<n/2$, $r=|T_n^{(k)}|\ge n/2$, and $l=r-m$. Since, as we have already observed, for a given k each tree contains at most one proper (k,l,m)-decomposition, we have

$$p_n(k,m) = \sum_{l=n/2-m}^{n-m-k} \hat{b}(n;k,l,m)\ , \tag{5}$$

where $\hat{b}(n;k,l,m)$ denote the probability that a random tree T_n contains a *proper* (k,l,m)-decomposition.

Let us look first at the terms of the sum (5) for which $l\ge m\log^2 n > n/4$. Since no tree of the forest $F(T_n^{(k)})$ is larger than $|T_n^{(k+1)}|=m$, and $T_n^{(k)}$ has $r=l+m$ vertices, the root of $T_n^{(k)}$ must have degree at least $r/m\ge\log^2 n$. Note now that if the algorithm GREEDY is applied to a random tree T_n, each tree on $r=l+m$ vertices is equally likely to appear as $T_n^{(k)}$. Furthermore, the probability that such a random tree on r vertices, where $n/2<r<n$, has the root of degree at least $\log^2 n$ is bounded from above by

$$\sum_{i=\lceil\log^2 n\rceil}^{r-1} \binom{r-1}{i} \frac{i(r-1)^{r-i-2}}{r^{r-2}} \le \sum_{i=\lceil\log^2 n\rceil}^{r-1} i\left(\frac{e}{i}\right)^i \le \exp(-\log^2 n)\ .$$

Thus, for $l\ge m\log^2 n$, we have

$$\hat{b}(n;k,l,m) \le n\exp(-\log^2 n),$$

and for n large enough

$$\sum_{l=m\log^2 n}^{n-m-k} \hat{b}(n;k,l,m) \le n^2\exp(-\log^2 n) \le n^{-3}\ . \tag{6}$$

Now let $n/4 < m + \sqrt{n} \le l \le m \log^2 n$. Then, the largest tree in the forest $F(T_n^{(k+1)})$ has at most $m \le r - m - \sqrt{n}$ vertices. Consequently, there exists s, $\sqrt{n} \le s \le l/3$, such that some components of $F(T_n^{(k)})$ have s vertices combined. Thus, for such a choice of l and m, from Stirling's formula we get

$$\begin{aligned}\frac{\hat{b}(n;k,l,m)}{\hat{a}(n;k,l,m)} &= \sum_{s=\lceil\sqrt{n}\rceil}^{\lfloor l/3\rfloor} \binom{l-1}{s-1} \frac{s^{s-2}(l-s-1)^{l-s-2}}{l^{l-2}} \\ &\le 3 \sum_{s=\lceil\sqrt{n}\rceil}^{\lfloor l/3\rfloor} \frac{(l-1)^{l-1/2}}{(s-1)^{s-1/2}(l-s)^{l-s+1/2}} \frac{s^{s-2}(l-s-1)^{l-s-2}}{l^{l-2}} \\ &\le 4 \sum_{s=\lceil\sqrt{n}\rceil}^{\lfloor l/3\rfloor} \left(\frac{l}{l-s}\right)^{5/2} \frac{1}{s^{3/2}} \le 30 n^{-1/4} \ .\end{aligned}$$

Note also that Fact 1 implies that, for some absolute constant C' and for each k and each $l, m \ge n/4 \log^2 n$, we have

$$a(n;k,l,m) \le C' n^{-5/2} \log^3 n \ . \tag{7}$$

Hence, for some constant C'',

$$\sum_{l=m+\sqrt{n}}^{m\log^2 n} \hat{b}(n;k,l,m) \le 30 n^{-1/4} \sum_{l=m+\sqrt{n}}^{m\log^2 n} \hat{a}(n;k,l,m) \le C'' n^{-5/3} \ . \tag{8}$$

Moreover, (7) implies that for n large enough

$$\sum_{l=m-\sqrt{n}}^{m+2\sqrt{n}} \hat{b}(n;k,l,m) \le \sum_{l=m-\sqrt{n}}^{m+2\sqrt{n}} \hat{a}(n;k,l,m) \le 3\sqrt{n}\, C' n^{-5/2} \log^3 n \le n^{-5/3} \ . \tag{9}$$

Finally, let $l \le m + 2\sqrt{n}$. Recall that $l + m = r \ge n/2$, so in this case we have $m \ge n/4 + \sqrt{n}$. Furthermore, since $l + m \ge n/2$ and $m \ge l$, for such a choice of l and m each (k,l,m)-decomposition must be proper, i.e., $\hat{b}(n;k,l,m) = \hat{a}(n;k,l,m)$. Consequently, using Fact 1, for $k = \lfloor x\sqrt{n} \rfloor$, $m = \lfloor yn \rfloor$, and $y \in (1/4, 1/2)$, we arrive at

$$\sum_{l=n/2-m}^{m-\sqrt{n}} \hat{b}(n;k,l,m) = \sum_{l=n/2-m}^{m-\sqrt{n}} \hat{a}(n;k,l,m) = \frac{1+o(1)}{n^{3/2}} f(x,y) \ , \tag{10}$$

where $f(x,y)$ is defined as in (4). Now the assertion follows from (5), (6), (8), (9), and (10). □

3. The limit distribution of H. In this section we shall use a certain type of self-similarity argument to find the limit distribution of H, the length of the path constructed by the algorithm GREEDY in a random tree T_n. Let us introduce first two sequences of auxiliary random variables $\{\hat{H}_i\}$ and $\{W_i\}$. We define H_i and W_i recursively, setting $\hat{H}_0 = \min_j\{|T_n^{(j)}| \le n/2\}$, $W_0 = |T_n^{(\hat{H}_0)}|$, while for $i \ge 1$,

$$\hat{H}_i = \min_j\{|T_n^{(j)}| \le W_{i-1}/2\}$$

and $W_i = |T_n^{(\hat{H}_i)}|$. Furthermore, we put $H_0 = \hat{H}_0$ and $H_i = \hat{H}_i - \hat{H}_{i-1}$ for $1 \le r \le n-1$. Thus, W_i denotes the size of the tree $T_n^{(k)}$ when it first drops under $W_{i-1}/2$, and H_i is the number of steps of the algorithm between two such moments. Note that for every $i \ge 0$, we have $W_i \le 2^{-i-1}n$.

Since the length of the path found by the algorithm can be written as a sum of H_i's, we have

$$\text{(11)} \quad \begin{aligned} \text{Prob}(H > k) &= \text{Prob}\left(\sum_{i\ge 0} H_i > k\right) \\ &= \text{Prob}(H_0 > k) + \sum_{j\ge 1} \text{Prob}\left(\sum_{i=0}^{j} H_i > k \wedge \sum_{i=0}^{j-1} H_i \le k\right). \end{aligned}$$

In order to characterize the behavior of the probabilities $\text{Prob}(\sum_{i=0}^{j} H_i > k \wedge \sum_{i=0}^{j-1} H_i \le k)$, we introduce an integral operator A, setting

$$(Ag)(x) = \int_0^x \int_{1/4}^{1/2} f(z,y) g((x-z)/\sqrt{y}) dy dz \,,$$

where f is the function defined by (4). Furthermore, for $x \ge 0$ let

$$\text{(12)} \qquad g_0(x) = \int_x^\infty \int_{1/4}^{1/2} f(z,y) dy dz,$$

and for $j \ge 1$,

$$\text{(13)} \qquad g_j = A g_{j-1} = A^j g_0 \,.$$

It is not hard to check that the integrals which appear in the definition of g_j converge. As a matter of fact, our next result provides an explicit upper bound for the value of g_j.

FACT 3. *For every $j \ge 0$, g_j is a nonnegative function, bounded from above by 1, such that*

$$\int_0^\infty g_j(x) dx \le \left(\int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(x,y) dy dx\right)^j < 2^{-j/2} \,.$$

Proof. Note first that $f(x,y)$ is related to the density of a random variable, so

$$\text{(14)} \qquad \int_0^\infty \int_{1/4}^{1/2} f(x,y) dy dx = 1 \,.$$

Thus, $g_0(x) \le 1$ for every $x \ge 0$. Furthermore, elementary computations show that

$$\text{(15)} \qquad \int_0^\infty g_0(x) dx = \int_0^\infty \int_{1/4}^{1/2} z f(z,y) dy dz = \sqrt{2/\pi} < 1 \,.$$

Now assume that the assertion holds for g_{j-1}, where $j \ge 1$. Then

$$g_j(x) = \int_0^x \int_{1/4}^{1/2} f(z,y) g_{j-1}((x-z)/\sqrt{y}) dy dz \le \int_0^x \int_{1/4}^{1/2} f(z,y) dy dz < 1 \,.$$

Moreover, we have

$$\int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(z,y)dydz = 1 - \frac{2\sqrt{2} + \ln(3+2\sqrt{2})}{\pi} < \frac{\sqrt{2}}{2}\,. \tag{16}$$

Thus,

$$\begin{aligned}\int_0^\infty g_j(x)dx &= \int_0^\infty \int_0^x \int_{1/4}^{1/2} f(z,y) g_{j-1}((x-z)/\sqrt{y})dydzdx \\ &= \int_0^\infty \int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(z,y) g_{j-1}(u)dydudz \\ &= \int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(z,y)dydz \int_0^\infty g_{j-1}(u)du \\ &< \sqrt{2}/2 \int_0^\infty g_{j-1}(u)du \le 2^{-j/2}\,. \qquad \square\end{aligned}$$

Our next result shows that the functions g_j are closely related to our problem.

LEMMA 4. *For every $x > 0$, we have*

$$\mathrm{Prob}(H_0 > \lfloor x\sqrt{n} \rfloor) = (1 + o(1))g_0,$$

and for $j \ge 1$,

$$\mathrm{Prob}\left(\sum_{i=0}^{j} H_i > \lfloor x\sqrt{n} \rfloor \wedge \sum_{i=0}^{j-1} H_i \le \lfloor x\sqrt{n} \rfloor\right) = (1 + o(1))g_i(x)\,,$$

where, for given positive constants c, C, the quantity $o(1)$ tends to 0 uniformly for $x \in (c, C)$.

Proof. We shall use the induction on j. The estimate for $\mathrm{Prob}(H_0 > \lfloor x\sqrt{n} \rfloor)$ follows immediately from Lemma 2 and (12). Moreover, for every k and $j \ge 1$,

$$\begin{aligned}&\mathrm{Prob}\left(\sum_{i=0}^{j} H_{\,i} > k \wedge \sum_{i=0}^{j-1} H_i \le k\right) \\ &\quad = \sum_{l=1}^{k-j} \mathrm{Prob}\left(H_0 = l \wedge \sum_{i=1}^{j} H_i > k - l \wedge \sum_{i=1}^{j-1} H_i \le k - l\right) \qquad (17) \\ &\quad = \sum_m \sum_l \mathrm{Prob}\left(\sum_{i=1}^{j} H_i > k - l \wedge \sum_{i=1}^{j-1} H_i \le k - l | H_0 = l \wedge W_0 = m\right) \\ &\qquad\qquad \times \mathrm{Prob}(H_0 = l \wedge W_0 = m).\end{aligned}$$

Now we use a simple "rescaling" idea. As we have already noticed in the proof of Lemma 2, in the first k steps the algorithm GREEDY employs no information about the tree $T^{(k)}$, except for its size. Thus, when the algorithm is run on a random tree, in each step we can treat $T_n^{(k)}$ as a random tree on $|T_n^{(k)}|$ vertices. Hence, $\mathrm{Prob}(\sum_{i=1}^{j} H_i > k - l \wedge \sum_{i=0}^{j-1} H_i \le k - l | H_0 = l \wedge W_0 = m)$ is precisely the probability that, if we apply the algorithm to a random rooted tree with m vertices, then $\sum_{i=0}^{j-1} H_i > k - l$ and $\sum_{i=0}^{j-2} H_i \le k - l$. Since, due to Lemma 2, the function $f(x, y)$

determines the joint distribution of (H_0, W_0), the assertion follows from (17) and the definition of A. □

As a straightforward consequence of Lemma 4 we obtain the following characterization of the limit distribution of H.

THEOREM 5. *For every constant $x \geq 0$,*

$$\lim_{n\to\infty} \text{Prob}(H > x\sqrt{n}) = h(x),$$

where

$$h(x) = \sum_{j=0}^{\infty} g_j(x) = \sum_{j=0}^{\infty} (A^j g_0)(x), \tag{18}$$

and functions g_j are defined by (12) *and* (13). *Equivalently, the function h is the only continuous solution of the integral equation*

$$\begin{aligned} h(x) &= g_0(x) + (Ah)(x) \\ &= \int_x^{\infty} \int_{1/4}^{1/2} f(z,y)dydz + \int_0^x \int_{1/4}^{1/2} f(z,y)h((x-z)/\sqrt{y})dydz\ , \end{aligned} \tag{19}$$

where the function f is given by (4).

Proof. Let us show first that equation (19) has a unique continuous solution. Indeed, due to Fact 3, the series $\sum_j g_j$ converges in the L_1-norm, and thus the function h defined by (18) is determined up to a set of measure zero. Clearly, for such a function, (19) holds, and since the kernel of this integral equation is absolutely continuous in the whole range of the integration, h can be chosen to be continuous.

Now let $\epsilon > 0$ be any positive constant. Choose $\delta > 0$ such that $h(x-\delta) \leq h(x) + \epsilon/3$, and pick J and N large enough so that the probability that the height of a random tree with less than $2^{-J-1}n$ vertices is larger than $\delta\sqrt{n}$ is smaller than $\epsilon/3$ for every $n \geq N$ (the existence of such constants follows from Theorem 0). Hence, since $W_J = |T_n^{(J)}| \leq 2^{-J-1}n$, the probability $P(\sum_{j\geq J} H_j \geq \delta\sqrt{n})$ that the algorithm will find a path of length larger than $\delta\sqrt{n}$ in $T_n^{(J)}$ is smaller than $\epsilon/3$. Moreover, from Lemma 4 and (11) it follows that one can uniformly approximate the probability $\text{Prob}(\sum_{j=0}^{J-1} H_j \geq x\sqrt{n})$ by $\sum_{j=0}^{J-1} g_j(x)$. Thus, for every $\epsilon > 0$ and n large enough,

$$\begin{aligned} h(x) - \epsilon &\leq \text{Prob}\left(\sum_{j=0}^{J-1} H_j > x\sqrt{n}\right) \leq \text{Prob}(H > x\sqrt{n}) \\ &\leq \text{Prob}\left(\sum_{j=0}^{J-1} H_j > (x-\delta)\sqrt{n}\right) + \epsilon/3 \leq h(x-\delta) + 2\epsilon/3 \leq h(x) + \epsilon\ , \end{aligned}$$

and the assertion follows. □

Remark. Let us note that once we know that the limit $\lim_{n\to\infty} \text{Prob}(H > x\sqrt{n})$ exists for each $x \geq 0$, the fact that h fulfills the integral equation $h = g_0 + Ah$ is quite natural and follows immediately from equation $H = H_0 + \lambda H$, where λ is an operator which plays the role of a "scaling factor."

4. The expectation of H. Once we have found the distribution of H, it is not hard to guess the value of its mean. Clearly, $EH/\sqrt{n}$ should converge to the

expected value of the random variable Z, where $P(Z > x) = h(x)$ and $h(x)$ is given by Theorem 5. But $xh(x) \to 0$ as $x \to \infty$ (in fact, Theorem 0 states that the probability that the height of a random tree is larger than x decreases exponentially with x), so

$$\mu = \mathrm{E}\, Z = \int_0^\infty h(x)dx.$$

Now if we integrate both sides of (19), after elementary calculations we arrive at

$$\mu = \int_0^\infty \int_{1/4}^{1/2} xf(x,y)dydx + \mu \int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(x,y)dydx\ , \tag{20}$$

so, consequently,

$$\mu = \frac{\int_0^\infty \int_{1/4}^{1/2} xf(x,y)dydx}{1 - \int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(x,y)dydx}\ .$$

Note also that, similarly to (19), equation (20) can easily be deduced from the "scaling" relation $H = H_0 + \lambda H$, once we know that the expectation of $H/\sqrt{n}$ exists. Unfortunately, the existence of the limit $\lim_{n\to\infty} \mathrm{E}\, H/\sqrt{n}$ is not implied by the existence of the limit distribution $h(x)$ (even if one can prove that the convergence is uniform for every $x \in (0,\infty)$ which indeed is the case—see section 5). Hence, we shall deduce (20) from Lemma 2, following the way which led us to Theorem 5.

We find first the limit distributions of random variables H_i. Not surprisingly, we shall do it recursively using an appropriate integral operator.

Thus, let B be the operator which maps an integrable function r into the function Br such that

$$(Br)(x) = \int_0^\infty \int_{1/4}^{1/2} \frac{f(z,y)}{\sqrt{y}} r\left(\frac{x}{\sqrt{y}}\right) dydz\ ,$$

where the function f is defined in (4).

Let us note two simple properties of B.

FACT 6. *For every nonnegative, integrable function r,*

$$\int_0^\infty (Br)(x)dx = \int_0^\infty r(x)dx\ .$$

Furthermore, if $m = \int_0^\infty xr(x)dx < \infty$, then

$$\int_0^\infty x(Br)(x)dx = m \int_0^\infty \int_{1/4}^{1/2} \sqrt{y} f(z,y)dydz < \frac{\sqrt{2}}{2} m < \infty\ . \tag{21}$$

Proof. Both equalities follows easily from the definition of B and (16). □

Now, for $x \geq 0$, let

$$r_0(x) = \int_{1/4}^{1/2} f(x,y)dy,$$

while for $j \geq 1$,

$$r_j = Br_{j-1}\ .$$

Then the distribution of H_j is characterized by the following local limit theorem.

LEMMA 7. *For every $j \geq 0$ and $x > 0$,*

$$\lim_{n\to\infty} \sqrt{n}\,\mathrm{Prob}(H_j = \lfloor xn \rfloor) = (1 + o(1))r_j(x)\ , \tag{22}$$

where for given positive constants c, C the quantity $o(1)$ tends to 0 uniformly for $x \in (c, C)$.

Proof. In the case when $j = 0$, (22) is an immediate consequence of Lemma 2. Now note that for $j \geq 1$ we have

$$\mathrm{Prob}(H_j = k_j) = \sum_k \sum_m \mathrm{Prob}(H_j = k_j | H_0 = k \wedge W_0 = m)\,\mathrm{Prob}(H_0 = k \wedge W_0 = m)\ .$$

But, similarly as in the proof of Lemma 4, $\mathrm{Prob}(H_j = k_j | H_0 = k \wedge W_0 = m)$ is just the probability that, for a random tree on m vertices, we have $H_{j-1} = k_j$. Thus, the assertion follows from Lemma 2. □

THEOREM 8.

$$\lim_{n\to\infty} \frac{\mathrm{E}\,H}{\sqrt{n}} = \mu\ ,$$

where

$$\mu = \sum_{j=0}^{\infty} \int_0^{\infty} x r_j(x)dx = \frac{\int_0^{\infty}\int_{1/4}^{1/2} x f(x,y)dydx}{1 - \int_0^{\infty}\int_{1/4}^{1/2} \sqrt{y} f(x,y)dydx} \tag{23}$$

$$= \frac{\sqrt{2\pi}}{2\sqrt{2} - \ln(3 + 2\sqrt{2})} = 2.352139\ldots\ .$$

Proof. Fix $\epsilon > 0$. Note first that, since $\sum_{j\geq J} H_j$ is the length of the path found by the algorithm in a random tree of size $W_j \leq 2^{-J-1}$, we can choose J such that the expectation of $\sum_{j\geq J} H_j$ is smaller than $0.1\epsilon\sqrt{n}$ for large n (by (2) we can take any J for which $\sqrt{2\pi}2^{-J/2-1/2} \leq 0.1\epsilon$). Furthermore, due to Fact 6, we can assume that J is large enough to have $\sum_{j\geq J}\int_0^{\infty} x r_j(x)dx \leq 0.1\epsilon$.

Now, for $j \geq 0$ and $b > a \geq 0$ define a random variable $H_j(a, b)$, setting

$$H_j(a,b) = \begin{cases} H_j & \text{if} \quad a\sqrt{n} < H_j < b\sqrt{n}, \\ 0 & \text{otherwise.} \end{cases}$$

Choose a constant C such that for all j, $0 \leq j \leq J$, and n large enough,

$$\mathrm{E}\,H_j(C, \infty)/\sqrt{n} \leq 0.1\epsilon/J,$$

and

$$\int_C^{\infty} x r_j(x)dx \leq 0.1\epsilon/J\ .$$

Note that such a constant C exists since, by (2),

$$\mathrm{E}\,H_j/\sqrt{n} \leq EH/\sqrt{n} \leq 3\ ,$$

and Fact 6 implies that $\int_0^{\infty} x r_j(x)dx < \infty$. Lemma 7 states that the function r_j approximates uniformly the distribution of H_j in every finite interval, so, for every j, $0 \leq j \leq J$, and n large enough we have

$$\left| \mathrm{E}\,H_j(0.1\epsilon/J, C)/\sqrt{n} - \int_{0.1\epsilon/J}^{C} x r_j(x)dx \right| \leq 0.3\epsilon/J\ .$$

Hence

$$\begin{aligned}\left|\mathrm{E}\,H\,/\sqrt{n}-\sum_{j\ge0}\int_0^\infty xr_j(x)dx\right| &\le \left|\sum_{j\ge0}\mathrm{E}\,H_j/\sqrt{n}-\sum_{j\ge0}\int_0^\infty xr_j(x)dx\right| \\ &\le \left|\sum_{j=0}^{J}\mathrm{E}\,H_j/\sqrt{n}-\sum_{j=0}^{J}\int_0^\infty xr_j(x)dx\right|+0.2\epsilon \\ \le \sum_{j=0}^{J}\left|\mathrm{E}\,H_j(0.1\epsilon/J,C)/\sqrt{n}-\int_{0.1\epsilon/J}^{C}xr_j(x)dx\right| &+\sum_{j=0}^{J}\mathrm{E}\,H(0,0.1\epsilon/J)/\sqrt{n} \\ +\sum_{j=0}^{J}\left(\mathrm{E}\,H(C,\infty)/\sqrt{n}+\int_0^{0.1\epsilon/J}xr_j(x)dx+\int_C^\infty xr_j(x)dx\right)&+0.2\epsilon<\epsilon.\end{aligned}$$

Thus, we have shown that the limit $\mu=\lim_{n\to\infty}(\mathrm{E}\,H/\sqrt{n})$ exists and is equal to $\sum_{j\ge0}\int_0^\infty xr_j(x)dx$. Moreover, from (21) we get

$$\begin{aligned}\mu&=\int_0^\infty xr_0(x)dx+\sum_{j=0}^{\infty}\int_0^\infty x(Br_j)(x)dx \\ &=\int_0^\infty\int_{1/4}^{1/2}xf(x,y)dydx+\mu\int_0^\infty\int_{1/4}^{1/2}\sqrt{y}f(x,y)dydx\ .\end{aligned}$$

Finally, the numerical values of two integrals which appear in the formula for μ are given by (15) and (16). □

5. Final remarks and comments. The main purpose of this note was to present a simple rescaling idea which allows us to "guess" and verify the asymptotic behavior of the algorithm without invoking generating functions. Thus, for the sake of simplicity, we have not stated our results in the strongest possible form. For instance, one can show that the local limit distribution of H, defined as

$$\hat{h}(x)=\lim_{n\to\infty}\sqrt{n}\,\mathrm{Prob}(H=\lfloor x\sqrt{n}\rfloor)\ ,$$

is the unique continuous solution of the integral equation

$$\hat{h}(x)=\int_{1/4}^{1/2}f(x,y)dy+\int_0^x\int_{1/4}^{1/2}\frac{f(z,y)}{\sqrt{y}}\hat{h}\left(\frac{x-z}{\sqrt{y}}\right)dydz\ ,$$

but the proof of the existence of $\hat{h}$ is slightly more involved than that for $h(x)$. It is also not hard to see that the convergence of $\mathrm{Prob}(H>x\sqrt{n})$ is uniform for $x\in(0,\infty)$, and that $\sqrt{n}\,\mathrm{Prob}(H=\lfloor x\sqrt{n}\rfloor)$ tends to $\hat{h}$ uniformly for $x\in(c,\infty)$, for every $c>0$.

We should also mention that many distribution results similar to Theorem 5 have been obtained by David Aldous, who ingeniously noticed that an appropriately scaled family of random trees converges to some compact stochastic object, the continuum random tree, whose properties can be studied using probabilistic tools (see [1]). Furthermore, because such a continuum approximation of a family of random trees implicitly takes care of all self-similarities inside it, all results which involve any type of rescaling arguments are typically easier to prove in this setting. Nonetheless, the combinatorial approach we have described, although probably not so elegant and compact, also has some advantages. First, it is rather elementary and obviates the use of stochastic processes. More importantly, one can apply it to compute moments of

some random variables, as well as to find local limit theorems for them, while in the "continuum" approach we lose the "local" information on a random tree when we approximate it by the stochastic continuum.

Let us also comment shortly on the performance of the algorithm GREEDY. Theorem 8 states that the expectation of the length H of the path found by GREEDY is only about 6.2% smaller than the expectation of the height $\tilde{H}$ of a random tree. The GREEDY procedure is also quite quick; if a random tree is given in, say, preordered form, it finds a path of length H in the expected time $O(H)$. However, we should keep in mind that the height of a vertex chosen at random from a random tree is $\sum_{j=2}^{n}(n)_j/n^j$ (see Meir and Moon [4]), which for large n, is just half of the expectation of the height of the tree $\tilde{H}$. Note also that the ratio $\mathrm{E}\,H/\,\mathrm{E}\,\tilde{H}$ is a rather crude measure of the efficiency of the algorithm; it would be much more informative to study directly the behavior of the random variable $H/\tilde{H}$. More specifically, we can ask what is the length of the path found by the algorithm in a tree $T_{n,h}$, chosen at random from the family of all rooted trees of n vertices and height $h = h(n)$. For large h, i.e., for $h/\sqrt{n} \to \infty$, the structure of $T_{n,h}$ was studied in [3]; we can show that then, with probability tending to 1 as $n \to \infty$, the algorithm constructs a path of length at least $h - O(n/h)$. However, the most interesting case, when h is of the order $\sqrt{n}$, seems to be much harder to handle.

Finally, let us mention that an analogous rescaling argument can be used to study the behavior of the algorithm GREEDY for different models of random trees as, for example, trees chosen uniformly at random from a simply generated family of trees. As a matter of fact, such an approach can be applied to any family of trees provided that

- every branch of a random tree of size m can be treated as a random tree of size m (so the self-similarity argument can be used);
- the formula for the number of forests which consist of such trees is known (so the scaling function f can be effectively computed).

Acknowledgments. I would like to express my deep gratitude to Boris Pittel who introduced this problem to me and made numerous insightful comments on earlier versions of the manuscript. I wish also to thank David Aldous for pointing out the connections between the results presented in the note and his continuum approach to random trees.

REFERENCES

[1] D. Aldous, *The continuum random tree II: An overview*, Stochastic Analysis, in Proc. Durham Symp. Stochastic Analysis 1990, London Math. Soc. Lecture Note Series, Vol. 167, M. T. Barlow and N. H. Bingham, eds., 1991, Cambridge University Press, Cambridge, pp. 23–70.

[2] P. Flajolet, J. Gao, A. M. Odlyzko, and B. Richmond, *The distribution of heights of binary trees and other simple trees*, Combin. Prob. Comput., 2 (1993), pp. 145–156.

[3] T. Łuczak, *The number of trees with a large diameter*, J. Austral. Math. Soc. Ser. A, 58 (1995), pp. 298–311.

[4] A. Meir and J. W. Moon, *On the altitude of nodes in random trees*, Canad. J. Math., 30 (1978), pp. 997–1015.

[5] A. Rényi and G. Szekeres, *On the height of trees*, J. Austral. Math. Soc., 7 (1967), pp. 497–507.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 330–339, May 1998

012

CIRCULANTS AND SEQUENCES*

KAREN L. COLLINS†

Abstract. A graph G is stable if its normalized chromatic difference sequence is equal to the normalized chromatic difference sequence of $G \times G$, the Cartesian product of G with itself. Let α be the independence number of G and let ω be its clique number. Suppose that G has n vertices. We show that the first ω terms of the normalized chromatic difference sequence of a stable graph G must be α/n and further show that if G has odd girth $2k+1$, then the first three terms of its normalized chromatic difference sequence are $\alpha/n, \alpha/n, \beta/n$, where $\beta \geq \alpha/k$. We derive from this sequence an upper bound on the independence ratio of G, which agrees with the lower bound of Häggkvist for $k = 2$ and of Albertson, Chan, and Haas for $k \geq 3$ [*Ann. Discrete Math.*, 13 (1982), pp. 89–100; *J. Graph Theory*, 17 (1993), pp. 581–588].

Zhou has shown that circulants and finite abelian Cayley graphs are stable. Let G be a circulant with symbol set S and n vertices [*Discrete Math.*, 90 (1991), pp. 297–311; *Discrete Appl. Math.*, 41 (1993), pp. 263–267]. We say that $S = \{a_1, a_2, \ldots, a_s\}$ is reversible if $a_1 + a_s = a_2 + a_{s-1} = \cdots = a_{\lfloor \frac{s}{2} \rfloor} + a_{\lceil \frac{s}{2} \rceil}$. We show that the independence ratio $\mu(G) \leq \mu(S)$ and that if S is reversible, then $\lim_{n\to\infty} \mu(G) = \mu(S)$. We conjecture that $\mu(G) = \mu(S)$ for a reversible circulant with sufficiently many vertices.

Key words. Cayley graph, chromatic difference sequence, circulant, graph homomorphism, independence ratio, no-homomorphism lemma, partitionable graph

AMS subject classification. 05C

PII. S0895480193252136

1. Introduction.

Let G be a graph. The chromatic difference sequence of G, $cds(G)$, is the sequence of positive integers of length equal to the chromatic number of G, with the ith term equal to the maximum number of vertices that can be additionally colored by using i instead of $i-1$ colors; see Albertson and Berman [1, 2]. The first appearance of this idea is Greene's and Kleitman's proof that comparability graphs have monotonically decreasing sequences (see [10, 11]); proofs from other perspectives and related works appear in [8, 9, 22, 24, 25]. In another direction, Stanley has developed a symmetric function generalization of the chromatic polynomial which contains the chromatic difference sequence; see [26, 27].

Let G have n vertices. The normalized chromatic difference sequence of G, $ncds(G)$, is $cds(G)$ with each term divided by n. This idea allows the chromatic sequences of graphs with different number of vertices to be fairly compared. For instance, the No-Homomorphism lemma in [4] proves that if $G \mapsto H$ homomorphically and H is vertex transitive, then $ncds(G)$ dominates $ncds(H)$. Hell and Nesetril therefore think about the graph homomorphism of G to H as coloring G with H [15]; see also [16, 17, 18, 20, 21]. Another generalization of the $ncds$ is Zhou's work in [30].

The first term of $ncds(G)$ is called the independence ratio, $\mu(G)$. See also [5, 13, 19, 31] for other connections with graph homomorphism. Häggkvist uses the No-Homomorphism lemma to prove that a graph G with odd girth at least 5 and minimum degree at least $(3/8)n$ maps homomorphically to the 5-cycle; hence $\mu(G) \geq 2/5$ [12]. Albertson, Chan, and Haas generalize this theorem to get that if G is a graph with odd girth $2k+1$ and minimum degree at least $(k/(2k+1))n$, then $\mu(G) \geq k/(2k+1)$

*Received by the editors July 14, 1993; accepted for publication (in revised form) May 8, 1997. http://www.siam.org/journals/sidma/11-2/25213.html

†Department of Mathematics, Wesleyan University, Middletown, CT 06459-0128 (kcollins@wesleyan.edu).

[3]. We show that if $ncds(G) = ncds(G \times G)$, then equality holds for each of these theorems. We also make a generalization to graphs with larger clique size.

A circulant G with n vertices is a vertex transitive graph with rotational symmetry such that two vertices are adjacent if their difference appears in a fixed set S. Define the size of S, $|S|$, to be the sum of the first and last elements of S, and $\mu(S)$ to be the independence ratio of any consecutive set of $|S|$ vertices in G. We show that $\mu(S) \geq \mu(G)$. Let a set T be reversible if $T = |T| - T$. Then we show $\lim_{n\to\infty} \mu(G) \geq \mu(T)$ whenever $S \subseteq T$. We also conjecture that the independence number of G with edges given by reversible T equals $\lfloor n \cdot \mu(T) \rfloor$ when n is sufficiently large. In light of Zhou's recent work [28, 29], it seems likely that these results may generalize to Cayley graphs of finite abelian groups. See also Larose, Laviolette, and Tardif [23].

Section 2 makes some useful definitions. In section 3 we prove an upper bound that we use throughout the paper, and describe some examples. Section 4 proves the independence ratio results that coincide with those of Häggkvist, and of Albertson, Chan, and Haas. Section 5 proves the further results on the independence ratio of circulants. We make some conjectures in section 6.

2. Definitions. All graphs will be simple and undirected. A circulant is a graph G with n vertices labeled $0, 1, 2, \ldots, n-1$ and edges determined by set $S \subseteq \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$. Vertex i is adjacent to j if $|i - j| \in S$ or $n - |i - j| \in S$. Circulant graphs are necessarily vertex transitive.

Let G and H be two graphs. Define the Cartesian product $G \times H$ to be the graph with vertex set $V(G) \times V(H)$. Two vertices (g_1, h_1) and (g_2, h_2) are adjacent if $g_1 = g_2$, and h_1 is adjacent to h_2 in H, or g_1 is adjacent to g_2 in G, and $h_1 = h_2$.

Define the chromatic number of graph G, called $\chi(G)$, to be the smallest integer n such that the vertices in G can be colored with n colors so that if two vertices are adjacent, then they receive different colors. Let $\chi(G) = m$. Define the chromatic sequence of G to be $\Delta_1, \Delta_2, \ldots, \Delta_m$, where Δ_i is equal to the number of vertices in the largest i-colorable vertex induced subgraph of G.

Let G be a graph. We define the chromatic difference sequence of G, $cds(G)$ to be $\alpha_1, \alpha_2, \ldots, \alpha_m$, where $\alpha_1 = \Delta_1$ and $\alpha_i = \Delta_i - \Delta_{i-1}$ for $2 \leq i \leq m$. Note that α_1 is the independence number of G. We will abbreviate $\alpha_1(G)$ as $\alpha(G)$. Define the independence ratio of G to be $\mu(G) = \alpha(G)/n$, where n is the number of vertices of G.

Let the number of vertices of G be n. Define the normalized chromatic difference sequence of G, $ncds(G)$, to be $\alpha_1/n, \alpha_2/n, \ldots, \alpha_m/n$. Then G is said to be stable if $ncds(G) = ncds(G^2)$. Define the ultimate chromatic difference sequence to be $NCDS(G) = \lim_{k\to\infty} ncds(G^k)$ [13, 30].

Note that the chromatic number of G^k is greater than or equal to the chromatic number of G, since G^k contains G as an induced subgraph. Conversely, $\chi(G) \geq \chi(G^k)$ by an easy argument. We let $f : V(G) \to \{1, 2, \ldots, \chi(G)\}$ be a coloring of G and

$$f(v_1, v_2, \ldots, v_k) = \sum_{i=1}^{n} f(v_i) \text{ (modulo } n).$$

If two vertices in G^k are adjacent, then they differ in only one position of their k-tuples, and hence must receive different colors modulo n.

3. An upper bound. A graph G is said to be stable if the normalized chromatic difference sequence of G is equal to the normalized chromatic difference sequence of $G \times G$, the Cartesian product of G with itself. Zhou has shown that circulants and Cayley graphs of finite abelian groups are stable; see [28, 29]. Siran has demonstrated the existence of Cayley graphs which are not stable [14]. See also Conjecture 1.

The generalized Petersen graph $P(7,3)$ in Figure 3.1(c) is a graph which is stable but neither a circulant nor the Cayley graph of a finite abelian group. We obtain an upper bound on $\alpha(G \times H)$ by using a clique cover of H and $cds(G)$. This proves that a graph G cannot be stable unless the first $\omega(G)$ terms of the $ncds(G)$ are equal.

Let $A = a_1 \geq a_2 \geq \cdots \geq a_s$ be a partition of n. Define its conjugate partition, A^*, by a_i^* equals the number of elements of A which are at least i, for $1 \leq i \leq a_1$.

THEOREM 3.1. *Let H be a graph, and let $C = c_1, c_2, \ldots, c_s$ be the sizes of a disjoint clique cover of H, where $c_1 \geq c_2 \geq \cdots \geq c_s$. Let C^* be the conjugate partition of C. For any graph G,*

$$\alpha_1(G \times H) \leq \sum_{i=1}^{\chi(G)} c_i^*(H)\alpha_i(G).$$

Proof. Let I be a maximum independent set in $G \times H$ and let I_h be the subset of I whose first entries are equal to h for each vertex h of H. Then I_h is isomorphic to an independent set in G. Clearly, if h_1 is adjacent to h_2 in H, $I_{h_1} \cap I_{h_2} = \emptyset$. Hence, if $h_1, h_2, \ldots, h_t$ form a clique in H, then $I_{h_1}, I_{h_2}, \ldots, I_{h_t}$ are t pairwise disjoint independent sets of G; hence $|I_{h_1} \cup I_{h_2} \cup \cdots \cup I_{h_t}| \leq \Delta_t = \alpha_1(G) + \alpha_2(G) + \cdots + \alpha_t(G)$, where $\alpha_j = 0$ if $j > \chi(G)$.

Thus $|I| \leq \sum_{i=1}^{s} \sum_{j=1}^{c_i(H)} \alpha_j(G)$. Each $\alpha_j(G)$ appears in the sum the same number of times as the number of cliques which have size at least j. This number is $c_j^*(H)$. □

COROLLARY 3.2. *Let H be a graph with $|H|$ vertices that contains at least one edge. Let G be a graph such that $\alpha_1(G) > \alpha_i(G)$ for some $2 \leq i \leq \omega(H)$. Then $\alpha(G \times H) < |H|\alpha(G)$.*

Proof. Let $C_1, C_2, \ldots, C_t$ be any disjoint clique covering of H that contains a clique of size $\omega(H)$. Since H contains an edge, $c_2^* > 0$. Note that $\sum_{i=1}^{s} c_i = |H| = \sum_{j=1}^{c_1} c_j^*$. Therefore $\sum_{i=1}^{l} c_i^*(H)\alpha_i(G) < |H|\alpha_1(G)$. □

COROLLARY 3.3. *Let G be a stable graph. Then*

$$\alpha_1(G) = \alpha_2(G) = \cdots = \alpha_{\omega(G)}(G).$$

The following is a direct proof that circulants satisfy Corollary 3.3.

THEOREM 3.4. *Let G be a circulant. Then $\alpha_1(G) = \alpha_2(G) = \cdots = \alpha_{\omega(G)}(G)$.*

Proof (direct). Let the vertices of G be numbered from 0 to $n-1$. Let I be a maximum independent set of G, and let W be a maximum clique of G. All addition is modulo n. Then $\{I + w | w \in W\}$ is a collection of ω disjoint maximum independent sets of G. Similarly, $\{W + i | i \in I\}$ is a collection of α disjoint maximum cliques. □

The Petersen graph (Figure 3.1(a)) is not a circulant, since $cds(P) = 4, 3, 3$. The $NCDS(P^k) = 1/3, 1/3, 1/3$; see [4]. The tree T in Figure 3.1(b) has $cds(T) = 4, 2$ and $ncds(T^2) = 1/2, 1/2$. It is easy to check that $P(7,3)$ in Figure 3.1(c) is not a circulant or the Cayley graph of a finite abelian group, and that $cds(G) = 5, 5, 4$. Label the outside vertices counterclockwise from the top as $1, 2, 3, 4, 5, 6, 7$, and label the vertex on the inside 7-cycle which is adjacent to i as i'. Let $T_1 = \{1, 3, 5\}$, $T_2 = \{2, 4\}$, $U_1 = \{2, 4\}$, and $U_2 = \{5, 6, 7\}$. All arithmetic is modulo 7. Then a maximum independent set in G^2 with 70 vertices is

$$\{(i,j) | j \in (T_1 + i - 1)\} \cup \{(i,j') | j \in (T_2 + i - 1)\} \cup \{(i',j) | j \in (U_1 + i - 1)\} \cup \{(i',j') | j \in (U_2 + i - 1)\}.$$

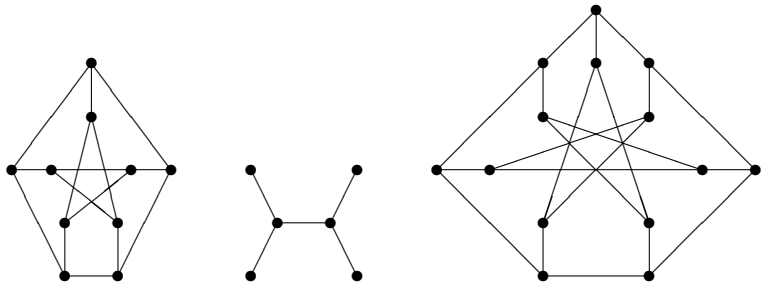


FIG. 3.1. *The Petersen graph in* (a) *is not stable; hence it is not a circulant. The tree T in* (b) *is not stable, but* T^2 *is stable. The generalized Petersen graph* $P(7,3)$ *in* (c) *is stable but is neither a circulant nor a finite abelian Cayley graph.*

A second maximum independent set is given by

$$\{(i,j)|j \in (T_1+i)\} \cup \{(i,j')|j \in (T_2+i)\} \cup \{(i',j)|j \in (U_1+i)\} \cup \{(i',j')|j \in (U_2+i)\}.$$

The graph G^2 is 3-colorable since G is, hence $ncds(G^2) = 5/14, 5/14, 2/7 = ncds(G)$, and G is stable.

4. Independence ratio of stable graphs. The *cds* of a stable graph G has its first ω terms equal to α. We prove that if G contains a partitionable graph H with independence number β and with the same clique size as G, then the next term in $cds(G)$ is at least α/β. Since odd cycles are partitionable, we apply this result to graphs with large odd girth. This gives an upper bound on the independence ratio which is the same number as the lower bounds of Häggkvist, and of Albertson, Chan, and Haas, when the minimum degree is bounded from below.

Define a graph G to be partitionable (or an (α,ω)-graph) if (i) $n = \alpha\omega + 1$; (ii) every vertex v is in exactly α independent sets of size α and ω cliques of size ω; (iii) the n independent sets of size α, say $S_1, S_2, \ldots, S_n$ and the n cliques of size ω, say $C_1, C_2, \ldots, C_n$, can be ordered so that $S_i \cap C_i = \emptyset$ and $S_i \cap C_j \neq \emptyset$ whenever $i \neq j$. Odd cycles and their complements are partitionable. Partitionable graphs are therefore related to the Strong Perfect Graph conjecture. See [6].

THEOREM 4.1. *Let G be a graph, and let H be a partitionable graph such that* $\omega(G) = \omega(H) = \omega$. *Then* $\alpha(G \times H) \leq \alpha(H) \sum_{i=1}^{\omega+1} \alpha_i(G)$.

Proof. Let $\alpha(H) = \beta$, $\alpha(G) = \alpha$. Then consider an independent set I in $G \times H$, where each vertex of H is replaced by a copy of G. Whenever two vertices in H are adjacent, then there is a matching between corresponding copies of G which joins isomorphic vertices of G. For every vertex v of H, let $I(v)$ be the intersection of I and the copy of G at v.

Thus if v_1 is adjacent to v_2 in H, then $I(v_1) \cap I(v_2) = \emptyset$, where $I(v_1), I(v_2)$ are considered as subsets of the vertices of G. If $v_1, v_2, \ldots, v_t$ is a clique in H, then $I(v_1), I(v_2), \ldots, I(v_t)$ is a collection of disjoint independent sets of G, that is, a partial coloring of G. Thus $|\cup_{j=1}^{t} I(v_j)| \leq \sum_{j=1}^{t} \alpha_j(G)$.

Label the vertices of H with 0 to $\beta\cdot\omega$ so that in $(H-0)$ the β disjoint ω-cliques are $C_1, C_2, \ldots, C_\beta$, where $m \in C_i$ exactly when $(i-1)\omega + 1 \leq m \leq i \cdot \omega$. Let $J(0) = I(0)$ and $J(k) = \bigcup(I(0) \cap I(j_1) \cap I(j_2) \cap \ldots I(j_k))$ where $(i-1)\omega + 1 \leq j_i \leq i\cdot\omega$ for $1 \leq i \leq k$. Each term in the union is nonempty if and only if the vertices $0, j_1, j_2, \ldots, j_k$ form an independent set in H. Notice that $J(k) \subseteq J(k-1)$.

Then we partition the vertices of I into β disjoint $(\omega+1)$-colorable subgraphs of G. Let $G_i = (J(i-1) - J(i)) \cup \bigcup_{r=1}^{\omega} I((i-1)\cdot\omega + r)$. Notice that $J(i) = \bigcup_{r=1}^{\omega} (I((i-1)\omega + r) \cap J(i-1))$; hence when $J(i)$ is subtracted from $J(i-1)$, we have removed $J(i-1) \cap \bigcup_{r=1}^{\omega} I((i-1)\omega + r)$ from $J(i-1)$. Also, $(i-1)\omega+1, (i-1)\omega+2, \ldots, i\cdot\omega$ forms a clique in H; hence G_i is the union of $\omega+1$ disjoint independent sets of G. Thus $|G_i| \le \sum_{s=1}^{\omega+1} \alpha_s(G)$.

Now the question remains of whether we have included every vertex of $I(0)$ in our partition. At each step i we include $J(i-1) - J(i)$ so that what we have remaining to include is $J(i)$. Thus after partitioning β times, one for each of the ω-cliques of H, we have remaining $J(\beta) = \bigcup(I(0) \cap I(j_1) \cap I(j_2) \cap \ldots I(j_\beta))$. But $J(\beta)$ must be empty, since $0, j_1, j_2, \ldots, j_\beta$ is a set of size $\beta+1$ and hence cannot be independent in H. Hence $|I| \le \sum_{i=1}^{\beta} |G_i| \le \beta(\sum_{i=1}^{\omega+1} \alpha_i(G))$. □

For example, if H is the 5-cycle, then $\alpha(G \times H) \le 2(\alpha_1(G) + \alpha_2(G) + \alpha_3(G))$.

The argument above can be applied separately to disjoint subgraphs of H to bound $\alpha(G \times H)$ further.

THEOREM 4.2. *Let G be a stable graph, and let H be an induced subgraph of G which is partitionable, and $\omega(G) = \omega(H)$. Let $\alpha(H) = \beta$, $\alpha(G) = \alpha$. Then*

$$\alpha_{\omega(G)+1}(G) \ge \alpha/\beta \quad \textit{and} \quad \beta/(\beta\omega+1) \ge \mu(G).$$

Proof. Suppose that I is a maximum independent set in G^2. Since G is stable, $\alpha(G^2) = |G|\alpha(G)$; hence if we consider G^2 as replacing each vertex of G with a copy of G, we must have that a maximum independent set in G^2 intersects each of the $|G|$ copies of G in $\alpha(G)$ vertices. Let $I(H)$ be I restricted to $H \times G$. Thus $|I(H)| = |H|\alpha(G)$. Now by Corollary 3.3, $\alpha = \alpha_1(G) = \alpha_2(G) = \cdots = \alpha_\omega(G)$. By the previous lemma, $|I(H)| = (\beta\omega+1)\alpha \le \beta(\omega\alpha + \alpha_{\omega+1})$, so $\alpha/\beta \le \alpha_{\omega+1}(G)$.

Let n be the number of vertices of G. Then $n \ge \sum_{i=1}^{\omega+1} \alpha_i(G) \ge \alpha(\omega + 1/\beta)$, so we get $\mu \le \beta/(\beta\omega+1)$. □

Define $\sigma(G)$ to be the size of the smallest chordless odd cycle of G. Let $\sigma(G) = 0$ if G has no chordless odd cycle, i.e., G is bipartite.

COROLLARY 4.3. *Let G be a stable graph with n vertices, and let $\sigma(G) \ge 2l+1$. Then $\alpha(G)/l \le \alpha_3(G)$ and $\mu(G) \le l/(2l+1)$.*

Proof. The first half of the proof follows from Theorem 4.2 and the fact that a $2l+1$ cycle is partitionable and has independence number l. □

This lower bound on $\mu(G)$ can be combined with the following results. Let $\delta(G)$ be equal to the minimum vertex degree of G.

THEOREM 4.4 (see Häggkvist [12]). *Let G be a graph with n vertices, $\sigma(G) \ge 5$, and $\delta > (3n)/8$. Then $\mu(G) \ge 2/5$.*

THEOREM 4.5 (see Albertson, Chan, and Haas [3]). *Let $l > 2$. Let G be a graph with n vertices, $\sigma(G) \ge 2l+1$, and $\delta > n/(l+1)$. Then $\mu(G) \ge l/(2l+1)$.*

COROLLARY 4.6. *Let G be a stable graph with $\sigma(G) \ge 5$ and $\delta(G) > 3n/8$. Then $\mu(G) = 2/5$.*

COROLLARY 4.7. *Let $l > 2$ and let G be a stable graph with $\sigma(G) \ge 2l+1$ and $\delta(G) > n/(l+1)$. Then $\mu(G) = l/(2l+1)$.*

In particular, any circulant G with n vertices and $\delta(G) > n/(l+1)$ satisfies $\alpha(G)/n = l/(2l+1)$. Since $gcd(l, 2l+1) = 1$, $2l+1$ must divide n. Thus if $2l+1$ does not divide n, G must have $\sigma(G) \le 2l-1$. If G is a circulant with $\delta(G) > 3n/8$ and 5 does not divide n, then G has a triangle.

Theorem 4.1 still holds if the graph H is replaced by the kth power of a cycle where the clique size does not divide the number of vertices. These circulants appear in Seymour's conjecture; see [7]. Let $W(m, l)$ be the circulant with m vertices and

$S = \{1, 2, 3, \ldots, l\}$ such that $m = j(l+1)+r$ and $r \neq 0$. Then $W(m,l)$ has $\omega = l+1$, $\alpha = j$, and $\chi = l+2$. When $r = 1$, $W(m,l)$ is partitionable.

THEOREM 4.8. *Let G be a graph. Let $m = j(l+1)+r$, where $r \neq 0$. Then $\alpha(W(m,l)) = j$ and $\alpha(G \times W(m,l)) \leq j \sum_{i=1}^{l+2} \alpha_i(G) + \sum_{i=1}^{r-1} \alpha_i(G)$.*

Proof. The proof is an easy generalization of the proof of Theorem 4.1. □

COROLLARY 4.9. *Let G be a stable graph that contains $W(m,l)$ such that $l+1$ does not divide m. Then $\alpha_{l+2}(G) \geq \alpha(G)/\alpha(W(m,l))$ and $\mu(G) \leq \alpha(W(m,l))/((l+1)j+1)$.*

5. Independence ratio of circulants. We prove that the independence ratio of circulant G with edge set given by S is less than or equal to the independence ratio of a graph $U(S)$ that depends only on S. This upper bound is therefore independent of the number of vertices in G. We then show that the limit of the independence ratio of a reversible circulant as the number of vertices goes to infinity equals the independence ratio of $U(S)$. We show two methods to embed a circulant which is not reversible into a circulant which is reversible, thus getting lower bounds for the limit of the independence ratio of any circulant.

Let $l \geq 2$. Let $S = \{a_1, a_2, \ldots, a_l\}$ be the edge set of circulant G. Let $|S| = a_1 + a_l$. Let $U(S)$ be the graph with vertices labeled $0, 1, 2, \ldots, (|S|-1)$ such that i is adjacent to j if $|i-j| \in S$. Then $U(S)$ is not a circulant because we are not including as edges the vertices i and j where $|i-j| \in |S| - S$. We abbreviate $\alpha(U(S))$ and $\omega(U(S))$ as $\alpha(S)$ and $\omega(S)$, respectively. Let $\mu(S) = \alpha(S)/|S|$. Then we get the following upper bound on the independence ratio. See also Conjecture 2.

THEOREM 5.1. *Let G be a circulant with n vertices and edge set given by S. Then $\mu(S) \geq \mu(G)$.*

Proof. Let $n = q|S| + r$ with $0 \leq r \leq |S| - 1$. Let I be a fixed maximum independent set of G and let $H_{j,k} = \{j, j+1, j+2, \ldots, j+k-1\}$ with arithmetic modulo n. Let i be the minimum of $|H(j,r) \cap I|$ over $0 \leq j \leq n-1$. If $i/r \leq \mu(S)$, we show $\mu(G) \leq \mu(S)$. Observe that any consecutive $|S|$ vertices of the circulant intersect with I in at most $\alpha(S)$ vertices. Therefore we can break up the circulant into q groups of $|S|$ and one group of r vertices, choosing the r vertices so that we achieve i as the minimum intersection with I. Then $\alpha(G) \leq q\alpha(S) + i \leq q\alpha(S) + r\alpha(S)/|S| = \frac{\alpha(S)}{|S|}(q|S| + r) = n\frac{\alpha(S)}{|S|}$. Hence $\mu(G) \leq \mu(S)$.

Suppose that $i/r > \alpha(S)/|S|$. Define $|S| = r_0$ and $r = r_1$. Let $r_{l+2} = r_{l+1}(\lfloor \frac{r_l}{r_{l+1}} \rfloor + 1) - r_l$ for nonnegative integer l. Let i_{l+2} be the minimum of $|H(j, r_{l+2} \cap I|$. Then we prove that $\mu(S) < i/r < i_2/r_2 < \cdots < i_L/r_L$, where r_L is the greatest common divisor of n and $|S|$. This gives the following contradiction: $i_L|S|/r_L > \alpha(S) \geq |H(j, |S|) \cap I| \geq i_L(|S|/r_L)$.

Now $gcd(n, |S|) = gcd(|S|, r)$ by the Euclidean algorithm. Since r_{l+2} is an integer linear combination of r_{l+1} and r_l, we have $gcd(r_l, r_{l+1}) = gcd(r_{l+1}, r_{l+2})$ for all nonnegative integers l. Let $r_l = q \cdot r_{l+1} + m$ with $0 \leq m \leq r_{l+1} - 1$. Then $r_{l+2} = r_{l+1}(q+1) - r_l = r_{l+1} - m$. Clearly $r_{l+2} = r_{l+1} - m < r_{l+1}$ unless $m = 0$, in which case r_{l+1} divides r_l and $r_{l+1} = gcd(n, |S|)$. Therefore the sequence $r_0, r_1, r_2, \ldots$ is a strictly decreasing sequence of positive integers with the same greatest common divisor, which must end in $r_L = gcd(n, |S|)$.

Assume by induction that $i_l/r_l < i_{l+1}/r_{l+1}$. Let $r_l = q \cdot r_{l+1} + m$. Fix j. Let $k_1 = |H(j + q \cdot r_{l+1}, m) \cap I|$ and $k_2 = |H(j + r_l, r_{l+2}) \cap I|$. Then $i_l \geq |H(j, r_l) \cap I| \geq q \cdot i_{l+1} + k_1$. Also, $k_1 + k_2 \geq i_{l+1}$. Hence $i_l - q \cdot i_{l+1} \geq i_{l+1} + k_2$. Since $i_l/r_l < i_{l+1}/r_{l+1}$, we get $i_{l+1}r_l - i_{l+1}q \cdot r_{l+1} > i_{l+1}r_{l+1} - k_2 r_{l+1}$. Simplifying, $k_2 r_{l+1} > i_{l+1}(r_{l+1} - m)$, but $r_{l+2} = r_{l+1} - m$. Therefore, $k_2/r_{l+2} > i_{l+1}/r_{l+1}$. This inequality must

hold for any value of j; hence it holds when k_2 is the minimum value, so $i_{l+2}/r_{l+2} > i_{l+1}/r_{l+1}$. □

THEOREM 5.2. *Let G be a circulant with n vertices and edge set given by S. Then $1/\omega(S) \geq \mu(G)$.*

Proof. For any circulant G, $n \geq \alpha(G)\omega(G)$ by Theorem 3.4; hence $1/\omega(G) \geq \mu(G)$. Let $S = \{a_1, a_2, \ldots, a_l\}$. Then $n \geq 2a_l$ by our definition of S. Thus $U(S)$ has $a_1 + a_l$ vertices and G has at least that many. We argue that $\omega(G) \geq \omega(S)$; hence $1/\omega(S) \geq 1/\omega(G) \geq \mu(G)$. Let the vertices of G be labeled $0, 1, 2, \ldots, n-1$. Then a copy of $U(S)$ is embedded in the subgraph of G induced by $0, 1, 2, \ldots, (a_1 + a_l - 1)$. Thus $\omega(G) \geq \omega(S)$. □

For any integer k, let $k - S = \{k - a_1, k - a_2, \ldots, k - a_l\}$. Define a set S to be reversible if $|S| - S = S$. Note that this means S is reversible if and only if $a_i + a_{l+1-i} = |S|$ for all $1 \leq i \leq l$. We show that a circulant G with n vertices and reversible set S maps homomorphically to the circulant H with $n - |S|$ vertices and the same set S.

LEMMA 5.3. *Let H be a reversible circulant with n vertices and with edge set given by S. Let $n > |S|$. Let G be the circulant with $n + |S|$ vertices and edge set given by S. Then G maps homomorphically to H.*

Proof. Let the vertices of H be $\{0, 1, 2, \ldots, n-1\}$ and the vertices of G be $\{0, 1, 2, \ldots, n + |S| - 1\}$. We define a homomorphism $f : G \to H$ by $f(i) = i$ if $0 \leq i \leq n-1$ and by $f(i+n) = i$ if $0 \leq i \leq |S| - 1$. Then we show that if v and u in G are adjacent, then $f(v)$ and $f(u)$ are adjacent in H.

Case 1. Suppose that $0 \leq v, u \leq n-1$. Then if v, u are adjacent in G, $|v - u|$ is in S or $n + |S| - S$. Every member of $n + |S| - S$ is at least n; hence $|v - u|$ is in S and, since $f(v) = v, f(u) = u$, we have that $f(v), f(u)$ are adjacent in H.

Case 2. Suppose that $0 \leq v, u \leq |S| - 1$. Then if $v + n, u + n$ are adjacent in G, $|(v + n) - (u + n)|$ is in S or $n + |S| - S$. But $|v - u| \leq |S| - 1$, hence $(v + n)$ and $(u + n)$ are adjacent in H.

Case 3. Suppose that $0 \leq v \leq n-1$ and $n \leq u + n \leq n + |S| - 1$, and $v, u + n$ are adjacent in G. Then $|u + n - v|$ is in S or $n + |S| - S$. If $u + n - v = a_j$ for some j, then $v - u = n - a_j$, so $v - u$ is in $n - S$. If $u + n - v = n + |S| - a_j$ for some j, then $u - v = |S| - a_j$, and since S is reversible, $u - v = a_{l+1-j}$. □

For any two graphs G and H for which G maps homomorphically to H, an i-colorable subgraph of H pulls back to an i-colorable subgraph of G. The No-Homomorphism lemma [4] is a special case of this fact. In the lemma below we show that in a reversible circulant, we can find a large independent set, which is based on a fixed small circulant with the same set of edges, by folding the large circulant onto the small one.

LEMMA 5.4. *Let H be a reversible circulant with n vertices and with edge set given by S. Let a_l be the largest element of S and let $n > |S| + a_l$. Let $G(k)$ be the circulant with $n + (k-1) \cdot |S|$ vertices and edge set given by S. Then $\alpha(G(k)) \geq \alpha(H) + (k-1)\alpha(S)$. Further, $\lim_{k\to\infty} \mu(G(k)) \geq \mu(S)$.*

Proof. Note that $G(1) = H$. By Lemma 5.3, $G(k+1)$ maps homomorphically to $G(k)$ for each positive integer k, and hence we have a homomorphism from $G(k+1)$ to H for every k. This map is $f_k : G(k) \to H$ by $f_k(i) = i$ for $0 \leq i \leq n-1$ and by $f_k(i + jn) = i$ for $0 \leq i \leq |S| - 1$ and $1 \leq j \leq k-2$. Any independent set in H can be pulled back to form an independent set in $G(k)$ since the preimage of a vertex in H is an independent set of vertices in $G(k)$. In particular we choose an independent set with as many vertices as possible chosen from $0, 1, 2, \ldots, |S| - 1$ in H. Since $n > |S| + a_l$, $n - a_l > |S|$ and there are no edges whose difference is in $n - S$

in the range of vertices $0, 1, 2, \ldots, |S| - 1$. Then $\alpha(G(k)) \geq \alpha(H) + (k-1) \cdot \alpha(S) \geq k\alpha(S)$. Hence the $\lim_{k\to\infty} \mu(G(k)) \geq \lim_{k\to\infty} k\alpha(S)/(n + (k-1) \cdot |S|) \geq \mu(S)$. Thus we get

$$\lim_{k\to\infty} \mu(G(k)) \geq \lim_{k\to\infty} \frac{k\alpha(S)}{(n + (k-1) \cdot |S|)} = \mu(S) \lim_{k\to\infty} \frac{k}{(n/|S| + k - 1)} = \mu(S). \qquad \square$$

Let $S = \{a_1, a_2, \ldots, a_l\}$ be a set which is not reversible. Then S can be embedded in a larger set which is reversible. This will add edges to the circulant which has S as its edge set, which may make the largest independent set smaller. Let $a_{l-1} + a_l = D$ and $a_1 + a_l = E$. Let $\hat{S} = S \cup (D - S)$ and $\tilde{S} = S \cup (E - S)$. Then it is easy to check that $\hat{S}, \tilde{S}$ are reversible and contain S. If S is reversible, define $\hat{S}, \tilde{S} = S$. Note that $\alpha(\hat{S}) \geq \alpha(S)$, since we have added only larger terms to S.

COROLLARY 5.5. *Let G be a circulant with n vertices and edge set given by $S = \{a_1, a_2, \ldots, a_l\}$. Define $L(S) = \lim_{n\to\infty} \mu(G)$. Then $\mu(S), 1/\omega(G) \geq L(S) \geq \mu(\hat{S}), \mu(\tilde{S})$. If G is reversible, then $L(S) = \mu(S)$.*

We apply these bounds to two special cases: a circulant with every element of S odd, and a circulant with S containing just two elements.

COROLLARY 5.6. *Let G be a circulant with n vertices and edge set given by S which contains only odd numbers. Then $L(S) = 1/2$.*

Proof. Since G is a circulant and $\omega(G) \geq 2$, $\alpha_1(G) = \alpha_2(G)$, hence $n \geq 2\alpha_1(G)$ and $\mu(G) \leq 1/2$. If n is even, then G is bipartite; hence $ncds(G) = 1/2, 1/2$. If n is odd, then G is not bipartite, but either S or $\hat{S}$ contains only odd numbers. Therefore $\alpha(\hat{S}) = |\hat{S}|/2$ by taking all vertices whose labels are even and less than $|\hat{S}|$. Thus $\mu(\hat{S}) = 1/2$ and Corollary 5.5 finishes the proof. $\square$

COROLLARY 5.7. *Let G be a circulant with n vertices and edge set given by $S = \{a, b\}$. Let $n > 2(a + b)$. Let q, r satisfy $b = qa + r$ with $0 \leq r \leq a - 1$. Then*

$$L(S) = \begin{cases} \dfrac{b + r}{2(a + b)} & q \text{ even}, \\ \dfrac{b + (a - r)}{2(a + b)} & q \text{ odd}. \end{cases}$$

Proof. We apply Corollary 5.5. Suppose that q is even. Then $\{i + 2ja | 0 \leq i \leq a-1, 0 \leq j \leq (q-2)/2\} \cup \{qa, qa+1, qa+2, \ldots, qa+r-1\}$ is an independent set in $U(S)$ and we can take $\alpha(S) \geq aq/2 + r = (b+r)/2$. If q is odd, then $\{i+2ja | 0 \leq i \leq a-1, 0 \leq j \leq (q-1)/2\}$ is an independent set in $U(S)$ and $\alpha(S) \geq a(q+1)/2 = (b + (a-r))/2$. Dividing by $|S| = a + b$ gives the result. $\square$

For example, $L(\{1, 2k\}) = k/(2k + 1)$. We provide some examples in Figure 5.1 below.

6. Conjectures. A circulant (or Cayley graph) G is not only stable, but all of its Cartesian powers G^k are stable. The graph in Figure 3.1(c) is not a circulant, but it is stable, which leads to the following conjecture.

CONJECTURE 1. *If G is stable, then G^k is stable for all positive integers k.*

Let G, H be reversible with edge set given by S as described in Lemma 5.4, and n is the number of vertices of H. Then $\mu(S) \geq \mu(G) \geq (\alpha(H) + (k-1)\alpha(S))/(n + (k-1)|S|)$. Hence $n \cdot \mu(S) + \alpha(S)(k-1) \geq \alpha(G) \geq \alpha(H) + \alpha(S)(k-1)$. We conjecture that for n as large as in Lemma 5.4, $\alpha(G)$ is always as large as possible. It is necessary that n be large; if H is the circulant with $S = \{1, 5\}$ and $n = 11$, then $\mu(S) = 1/2$, but $\mu(H) = 3/11$, and $11/2$ is much greater than 3.

S	Best lower bound	Best upper bound
$1,2,4$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 1/3$
$1,2,5$	$1/\omega(S) = 1/3$	$\mu(\tilde{S}) = 1/3$
$1,2,6$	$\mu(S) = 2/7$	$\mu(\tilde{S}) = 2/7$
$1,2,7$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 1/3$
$1,3,4$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 2/7$
$1,3,6$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 1/3$
$1,4,5$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 1/3$
$1,4,6$	$\mu(S) = 3/7$	$\mu(\hat{S}) = 2/5$
$1,4,7$	$\mu(S) = 3/8$	$\mu(\hat{S}) = 3/8$
$1,5,6$	$1/\omega(S) = 1/3$	$\mu(\tilde{S}) = 2/7$
$1,6,7$	$1/\omega(S) = 1/3$	$\mu(\hat{S}) = 4/13$

FIG. 5.1. *These are the best upper and lower bounds for $L(S)$ for some small values of S. See also Conjecture* 3.

CONJECTURE 2. *Let G be a reversible circulant with n vertices and edge set given by $S = \{a_1, a_2, \ldots, a_l\}$. If $n > |S| + a_l$, then $\alpha(G) = \lfloor n \cdot \mu(S) \rfloor$.*

By Corollary 5.5, we have $L(S) \geq \mu(\hat{S}), \mu(\tilde{S})$; Figure 5.1 shows that $L(\{1,2,4\}) = \mu(\hat{S})$, and $L(\{1,2,5\}) = \mu(\tilde{S})$.

CONJECTURE 3. $L(S) = \max\{\mu(\hat{S}), \mu(\tilde{S})\}$.

REFERENCES

[1] M. O. ALBERTSON AND D. M. BERMAN, *The chromatic difference sequence of a graph*, J. Combin. Theory Ser. B, 29 (1980), pp. 1–12.

[2] M. O. ALBERTSON AND D. M. BERMAN, *Critical graphs for chromatic difference sequences*, Discrete Math., 31 (1980), pp. 225–233.

[3] M. O. ALBERTSON, L. CHAN, AND R. HAAS, *Independence and graph homomorphisms*, J. Graph Theory, 17 (1993), pp. 581–588.

[4] M. O. ALBERTSON AND K. L. COLLINS, *Homomorphisms of 3-chromatic graphs*, Discrete Math., 54 (1985), pp. 127–132.

[5] J. A. BONDY AND P. HELL, *A note on the star chromatic number*, J. Graph Theory, 14 (1990), pp. 479–482.

[6] V. CHVÁTAL, R. L. GRAHAM, A. F. PEROLD, AND S. H. WHITESIDES, *Combinatorial designs related to the strong perfect graph conjecture*, Discrete Math., 26 (1979), pp. 83–92.

[7] R. J. FAUDREE, R. J. GOULD, M. S. JACOBSON, AND R. H. SCHELP, *Seymour's conjecture*, in Advances in Graph Theory, V. R. Kulli, ed., Vishwa International Pub., Gulbarga, 1991, pp. 163–171.

[8] S. V. FOMIN, *Finite partially ordered sets and Young tableaux*, Soviet Math. Dokl., 19 (1978), pp. 1510–1514 (in English).

[9] S. V. FOMIN, *A duality theorem for partially ordered sets: Algorithms*, in Mathematical Methods for Constructing and Analyzing Algorithms 237, Nauka, Leningrad, 1990, pp. 190–199.

[10] C. GREENE AND D. G. KLEITMAN, *The structure of Sperner k-families*, J. Combin. Theory Ser. A, 20 (1976), pp. 41–68.

[11] C. GREENE, *Some partitions associated with a partially ordered set*, J. Combin. Theory Ser. A, 20 (1976), pp. 69–79.

[12] R. HÄGGKVIST, *Odd cycles of specified lengths in non-bipartite graphs*, Ann. Discrete Math., 13 (1982), pp. 89–100.

[13] G. HAHN, P. HELL, AND S. POLJAK, *On the ultimate independence ratio of a graph*, European J. Combin., 16 (1995), pp. 253–261.

[14] G. HAHN AND J. SIRAN, *A note on intersecting cliques in Cayley graphs*, J. Combin. Math. Combin. Comput., 18 (1995), pp. 57–63.

[15] P. HELL AND J. NESETRIL, *On the complexity of H-coloring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.

[16] P. Hell, J. Nesetril, and X. Zhu, *Complexity of tree homomorphisms*, Discrete Appl. Math., 70 (1996), pp. 23–36.
[17] P. Hell, J. Nesetril, and X. Zhu, *Duality and polynomial testing of tree homomorphisms*, Trans. Amer. Math. Soc., 348 (1996), pp. 1281–1297.
[18] P. Hell, J. Nesetril, and X. Zhu, *Duality of graph homomorphisms*, in Combinatorics: Paul Erdős is Eighty, Bolyai Society Mathematical Studies, vol. 2, János Bolyai Math. Soc., Budapest, 1996, pp. 271–282.
[19] P. Hell, X. Yu, and H. Zhou, *Independence of graph powers*, Discrete Math., 127 (1994), pp. 213–220.
[20] P. Hell and X. Zhu, *The existence of homomorphisms to oriented cycles*, SIAM J. Discrete Math., 8 (1995), pp. 208–222.
[21] P. Hell and X. Zhu, *Homomorphisms to oriented paths*, Discrete Math., 132 (1994) pp. 107–114.
[22] A. Hoffman and D. Schwartz, *On partitions of a partially ordered set*, J. Combin. Theory Ser. B, 23 (1977), pp. 3–13.
[23] B. Larose, F. Laviolette, and C. Tardif, *On normal Cayley graphs and hom-idempotent graphs*, European J. Combin., to appear.
[24] M. Saks, *A short proof of the existence of k-saturated partitions of a partially ordered set*, Adv. Math., 33 (1979), pp. 207–211.
[25] M. Saks, *Some sequences associated with combinatorial structures*, Discrete Math., 59 (1986), pp. 135–166.
[26] R. Stanley, *A symmetric function generalization of the chromatic polynomial of a graph*, Adv. Math., 111 (1995), pp. 166–194.
[27] R. Stanley, *Graph colorings and related symmetric functions: Ideas and applications*, Discrete Math., to appear.
[28] H. Zhou, *The chromatic difference sequence of the Cartesian product of graphs*, Discrete Math., 90 (1991), pp. 297–311.
[29] H. Zhou, *The chromatic difference sequence of the Cartesian product of graphs, Part* II, Discrete Appl. Math., 41 (1993), pp. 263–267.
[30] H. Zhou, *On the ultimate normalized chromatic difference sequences of graphs*, Discrete Math., 148 (1996), pp. 287–297.
[31] X. Zhu, *On the bounds for the ultimate independence ratio of a graph*, Discrete Math., 156 (1996), pp. 229–236.

SIAM J. DISCRETE MATH.
Vol. 11, No. 2, pp. 340–346, May 1998

013

SEMIKERNELS AND (k,l)-KERNELS IN DIGRAPHS*

H. GALEANA-SÁNCHEZ† AND XUELIANG LI‡

Abstract. Let D be a digraph with minimum indegree at least one. The following results are proved: a digraph D has a semikernel if and only if its line digraph $L(D)$ does; the number of $(k,1)$-kernels in $L(D)$ is less than or equal to that in D; if the number of (k,l)-kernels in D is less than or equal to the number of $(2,l)$-kernels in $L(D)$, and if $L(D)$ has a (k,l)-kernel, then D has a (k',l')-kernel for $k'+l \leq k$, $l \leq l'$. As a consequence, it obtains previous results about kernels and quasikernels in the line digraph.

It is also proved that any digraph has a (k,l)-kernel with $l \geq 2k-2$, $k \geq 1$, generalizing a previous result on the existence of quasikernels in digraphs.

Key words. kernel, (k,l)-kernels, line digraph, semikernels

AMS subject classification. 0C20

PII. S0895480195291370

1. Introduction.

For general concepts we refer the reader to [1].

DEFINITION 1.1. *Let $D=(V(D),A(D))$ be a digraph. The line digraph $L(D)$ of D is the digraph $L(D)=(V(L(D)),A(L(D)))$ with set of vertices the set of arcs of D, and for any $h,k \in A(D)$ there is $(h,k) \in A(L(D))$ if and only if the corresponding arcs h,k induce a directed walk in D, i.e., the terminal endpoint of h is the initial endpoint of k. In what follows we denote the arc $h=(u,v) \in A(D)$ and the vertex $h \in V(L(D))$ by the same symbol. If H is a set of arcs in D, it is also a set of vertices of $L(D)$. When we want to emphasize our interest in $H \subseteq A(D)$ as a set of vertices of $L(D)$, we use the symbol H_L instead of H.*

DEFINITION 1.2. *A set $K \subseteq V(D)$ is said to be a kernel if it is both independent (a vertex in K has no successor in K) and absorbing (a vertex not in K has a successor in K).*

This concept was introduced by Von Neumann [11] and it has found many applications [1], [2]. Several authors have been investigating sufficient conditions for the existence of kernels in digraphs, namely, Von Neumann and Morgenstern [11] , Richardson [13], Duchet and Meyniel [4], [5], and Galeana-Sánchez and Neumann-Lara [7]. In [9], Harminc considered the existence of kernels in the line digraph of a given digraph D and he proved the following result.

THEOREM 1.1 (see [8]). *The number of kernels of a digraph D is equal to the number of kernels in its line digraph $L(D)$.*

DEFINITION 1.3 (see [12]). *A semikernel S of D is an independent set of vertices such that, for every $z \in (V(D) \setminus S)$ for which there exists an Sz-arc, there also exists a zS-arc.*

The concept of semikernel is nearly related to that of kernel, and is very useful to find kernels in digraphs, where every induced subdigraph of a digraph D has a semikernel then D has a kernel (see [12]). In [8] it was proved that the number of

*Received by the editors September 5, 1995; accepted for publication (in revised form) May 8, 1997.

http://www.siam.org/journals/sidma/11-2/29137.html

†Instituto de Matemáticas, Universidad Nacional Autónoma de México, México, D.F., C.P. 04510, México (imate@redvaxl.dgsca.unam.mx).

‡Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, People's Republic of China. The research of this author was supported by the Third World Academy of Sciences.

semikernels of a digraph D is less than or equal to the number of semikernels of $L(D)$. In this paper we prove that a digraph D has a semikernel if and only if $L(D)$ does.

DEFINITION 1.4. *A quasikernel Q of a digraph D is an independent set of vertices such that $V(D) = Q \cup \Gamma^-(Q) \cup \Gamma^-(\Gamma^-(Q))$ (where for any $A \subseteq X$, $\Gamma^-(A) = \{x \in X | x$ has a successor in $A\}$).*

In [3], Chvátal and Lovász proved that any digraph has a quasikernel; a generalization of this result was obtained by Duchet, Hamidoune, and Meyniel [6]. In [8] the following result was proved.

THEOREM 1.2 (see [8]). *If D is a digraph such that every vertex has indegree at least one, then the number of quasikernels of D is less than or equal to the number of quasikernels of its line digraph $L(D)$.*

DEFINITION 1.5. *Let D be a digraph. By the directed distance $d_D(x,y)$ from the vertex x to the vertex y in D we mean the length of a shortest directed path from x to y in D.*

DEFINITION 1.6 (see [10]). *Let k and l be natural numbers with $k \geq 2$, $l \geq 1$. A set $J \subseteq V(D)$ will be called a (k,l)-kernel of the digraph D if*

(1) *for each $x' \neq x$, $\{x, x'\} \subseteq J$ we have $d_D(x,x') \geq k$,*

(2) *for each $y \in (V(D) \setminus J)$, there exists $x \in J$ such that $d_D(y,x) \leq l$.*

Notice that, for $k = 2$, $l = 1$, we have that a (k,l)-kernel is a kernel and that for $k = 2$, $l = 2$, a (k,l)-kernel is a quasikernel.

2. Semikernels and (k,l)-kernels in the line digraph.

DEFINITION 2.1 (see [9]). *Let $D = (V(D), A(D))$ be a digraph. We denote by $\mathcal{P}(X)$ the set of all the subsets of the set X, and $f : \mathcal{P}(V(D)) \to \mathcal{P}(A(D))$ will denote the function defined as follows: for each $Z \subseteq V(D)$, $f(Z) = \{(u,x) \in A(D) | x \in Z\}$. Also, we denote by $\bar{f} : \mathcal{P}(A(D)) \to \mathcal{P}(V(D))$ the function defined as follows: for each $A \subseteq A(D)$, $\bar{f}(A) = \{x \in V(D) | (u,x) \in A\}$.*

LEMMA 2.1 (see [9]). *If $Z \subseteq V(D)$ is an independent set of D, then $f(Z)_L$ is an independent set in $L(D)$.*

THEOREM 2.1. *If D is a digraph such that every vertex has indegree at least one, then D has a semikernel if and only if $L(D)$ has a semikernel.*

Proof. If D has a semikernel S, then from the proof of Theorem 2.1 [8], we know that $f(S)_L$ is a semikernel of $L(D)$.

Conversely, if $L(D)$ has a semikernel A, then we will show that $\bar{f}(A)$ is a semikernel of D.

First we prove that $\bar{f}(A)$ is independent. By contradiction, if $\bar{f}(A)$ is not independent, then there are two vertices $x, y \in \bar{f}(A)$ such that $(x,y) \in A(D)$. Since $x \in \bar{f}(A)$, there exists a vertex $u \in V(D)$ such that $(u,x) \in A$. Since $((u,x),(x,y))$ is an $A(x,y)$-arc in $L(D)$ and A is a semikernel of $L(D)$, there must be an arc $(y,v) \in A(D)$ such that $(y,v) \in A$ and $((x,y),(y,v)) \in A(L(D))$. Since $y \in \bar{f}(A)$, there is a $t \in V(D)$ such that $(t,y) \in A$. Then we have $\{(t,y),(y,v)\} \subseteq A$, with $((t,y),(y,v)) \in A(L(D))$, which contradicts the independence of A. We conclude that $\bar{f}(A)$ is independent.

Now, let $y \in V(D)$ such that there is a $\bar{f}(A)y$-arc; there exists $x \in \bar{f}(A)$ with $(x,y) \in A(D)$. Since $x \in \bar{f}(A)$, there is an arc $(z,x) \in A$. Thus $((z,x),(x,y))$ is an $A(x,y)$-arc in $L(D)$. Since A is a semikernel of $L(D)$, there exists an $(x,y)A$-arc in $L(D)$. Let that arc be $((x,y),(y,u))$ so that $(y,u) \in A$ and then $u \in \bar{f}(A)$. We have proved that there is a $y\bar{f}(A)$-arc in D. Hence $\bar{f}(A)$ is a semikernel of D. □

THEOREM 2.2. *Let D be a digraph such that each vertex has indegree at least one. Then the number of $(k,1)$-kernels in $L(D)$ is less than or equal to the number of $(k,1)$-kernels in D.*

Proof. First we will prove that if $\bar{K}$ is a $(k,1)$-kernel of $L(D)$, then $\bar{f}(\bar{K})$ is a $(k,1)$-kernel of D.

Let $\bar{K}$ be a $(k,1)$-kernel of $L(D)$.

(a) If $x \neq x'$, $\{x,x'\} \subseteq \bar{f}(\bar{K})$, then $d_D(x,x') \geq k$.

By contradiction, suppose that $d_D(x,x') = n < k$. Take $\alpha = (x = x_0, x_1, \ldots, x_n = x')$, a shortest directed path from x to x' contained in D. Since $x \in \bar{f}(\bar{K})$, there exists $u \in V(D)$ such that $(u,x) \in \bar{K}$. Denote by $a_i = (x_{i-1}, x_i) \in A(D)$, $1 \leq i \leq n$, and consider the following two possibilities:

If $a_n = (x_{n-1}, x_n) \in \bar{K}$, consider that $((u,x), a_1, a_2, \ldots, a_n)$ is a directed path from (u,x) to a_n contained in $L(D)$ of length $n < k$ with $\{(u,x), a_n\} \subseteq \bar{K}$; this contradicts part (1) of Definition 1.6, as $\bar{K}$ is a $(k,1)$-kernel of $L(D)$.

If $a_n = (x_{n-1}, x_n) \notin \bar{K}$, then it follows from part (2) of Definition 1.6 that there exists $(x_n, z) \in \bar{K}$ such that $((x_{n-1}, x_n), (x_n, z)) \in A(L(D))$ (as $\bar{K}$ is a $(k,1)$-kernel of $L(D)$). On the other hand, $x' = x_n \in \bar{f}(\bar{K})$, so there exists $v \in V(D)$ with $(v, x_n) \in \bar{K}$ and then $((v,x_n),(x_n,z)) \in A(L(D))$ with $\{(v,x_n),(x_n,z)\} \subseteq \bar{K}$, contradicting part (1) of Definition 1.6 as $\bar{K}$ is a $(k,1)$-kernel of $L(D)$, $k \geq 2$.

(b) If $y \in V(D) \setminus \bar{f}(\bar{K})$, then there exists $x \in \bar{f}(\bar{K})$ such that $(y,x) \in A(D)$.

Since $y \in V(D)$, it follows from the hypothesis of Theorem 2.1 that there exists $u \in V(D)$ with $(u,y) \in A(D)$. Now $y \in V(D) \setminus \bar{f}(\bar{K})$ implies $(u,y) \in V(L(D)) \setminus \bar{K}$; it follows from part (2) of Definition 1.6 that there exists $(y,x) \in \bar{K}$ such that $((u,y),(y,x)) \in A(L(D))$ (because $\bar{K}$ is a $(k,1)$-kernel of $L(D)$). Since $(y,x) \in \bar{K}$, we have $x \in \bar{f}(\bar{K})$ and (b) is proved.

Let $\mathcal{K}_1$ be the set of all $(k,1)$-kernels of $L(D)$ and $\mathcal{K}$ the set of all $(k,1)$-kernels of D. We will prove that $\bar{f}' : \mathcal{K}_1 \to \mathcal{K}$, where $\bar{f}'$ is the restriction of $\bar{f}$ to $\mathcal{K}_1$, is an injective function.

(c) If $\bar{K}_1, \bar{K}_2 \in \mathcal{K}_1$, $\bar{K}_1 \neq \bar{K}_2$, then $\bar{f}'(\bar{K}_1) \neq \bar{f}'(\bar{K}_2)$.

Suppose, without loss of generality, that $\bar{K}_1 \setminus \bar{K}_2 \neq \emptyset$ and take $(u,v) \in \bar{K}_1 \setminus \bar{K}_2$. Clearly, from Definition 2.1 $v \in \bar{f}'(\bar{K}_1)$ and we will show that $v \notin \bar{f}'(\bar{K}_2)$. By contradiction, assume $v \in \bar{f}'(\bar{K}_2)$; hence there exists $(z,v) \in \bar{K}_2$. Since $(u,v) \notin \bar{K}_2$, it follows from part (2) of Definition 1.6 that there exists $(v,y) \in \bar{K}_2$. Hence $((z,v),(v,y)) \in A(L(D))$ with $\{(z,v),(v,y)\} \subseteq \bar{K}_2$, contradicting part (1) of Definition 1.6, because $\bar{K}_2$ is a $(k,1)$-kernel of $L(D)$. We conclude that $v \notin \bar{f}'(\bar{K}_2)$, and so $\bar{f}'(\bar{K}_1) \neq \bar{f}'(\bar{K}_2)$ and $\bar{f}'$ is injective. □

Remark 2.1. The hypothesis that each vertex has indegree at least one cannot be omitted in Theorem 2.2 for $k \geq 3$. It suffices to consider D with $V(D) = \{u_1,u_2,u_3,u_4,u_5,u_6\}$ and $A(D) = \{(u_1,u_2),(u_2,u_3),(u_4,u_5),(u_5,u_6)\}$. Here D has no $(k,1)$-kernel but $L(D)$ has one $(k,1)$-kernel for any $k \geq 3$.

Remark 2.2. The inequality announced in Theorem 2.2 can be strict for $k \geq 3$. Consider D with $V(D) = \{u_1,u_2,u_3\}$ and $A(D) = \{(u_1,u_2),(u_2,u_3),(u_3,u_1),(u_1,u_3)\}$. Then D has a $(k,1)$-kernel and $L(D)$ does not have any $(k,1)$-kernel for $k \geq 3$.

THEOREM 2.3. *Let D be a digraph such that every vertex has indegree at least one. Then the number of (k,l)-kernels in D is less than or equal to the number of $(2,l)$-kernels in $L(D)$.*

Proof. First we will prove that if K is a (k,l)-kernel of D, $k \geq 2$, then $f(K)$ is a $(2,l)$-kernel of $L(D)$.

Let K be a (k,l)-kernel of D.

(a) If $a \neq a'$, $\{a,a'\} \subseteq f(K)$, then $d_{L(D)}(a,a') \geq 2$.

By contradiction, suppose that $d_{L(D)}(a,a') \leq 1$, as $a \neq a'$, then $d_{L(D)}(a,a') = 1$; it follows from Definition 1.1 that the terminal endpoint of a is the initial endpoint

of a'. Denoting $a = (x,y), a' = (y,z)$, it follows from Definition 2.1 and the fact $\{a,a'\} \subseteq f(K)$ that $\{y,z\} \subseteq K$, so $(y,z) \in A(D)$ with $\{y,z\} \subseteq K$, contradicting part (1) of Definition 1.6 as K is a (k,l)-kernel of D.

(b) If $b \in V(L(D)) \setminus f(K)$, then there exists $a \in f(K)$ such that $d_{L(D)}(b,a) \leq l$.

Denoting $b = (u,v)$ we have from Definition 2.1 and the fact $b \notin f(K)$ that $v \notin K$; now part (2) of Definition 1.6 implies that there exists $w \in K$ such that $d_D(v,w) = n \leq l$. Let $(v = x_0, x_1, \ldots, x_n = w)$ be a shortest directed path from v to w in D and denote $a_i = (x_{i-1}, x_i) \in A(D)$. Then $(b, a_1, a_2, \ldots, a_n)$ is a directed path in $L(D)$ of length n from b to a_n, and since $w \in K$ we have $a_n \in f(K)$, so taking $a = a_n$, (b) is proved.

Let $\mathcal{K}$ be the set of all (k,l)-kernels of D, $k \geq 2$, and let $\mathcal{K}_2$ be the set of all $(2,l)$-kernels of $L(D)$. We will prove that $f' : \mathcal{K} \to \mathcal{K}_2$, where f' is the restriction of f to $\mathcal{K}$, is an injective function.

(c) If $K_1, K_2 \in \mathcal{K}$, $K_1 \neq K_2$, then $f'(K_1) \neq f'(K_2)$.

Suppose, without loss of generality, that $K_1 \setminus K_2 \neq \emptyset$ and take $v \in K_1 \setminus K_2$. It follows from the hypothesis of Theorem 2.3 that there exists $(u,v) \in A(D)$; it follows from Definition 2.1 that $(u,v) \in f'(K_1) \setminus f'(K_2)$ and so $f'(K_1) \neq f'(K_2)$. □

Remark 2.3. The hypothesis that each vertex has indegree at least one cannot be omitted in Theorem 2.3 for $l \geq 2$. Consider that $D \cong T_2$ is the directed path of length two; $L(D) \cong T_1$ is the directed path of length one, D has two $(2,l)$-kernels for any $l \geq 2$, and $L(D)$ has just one $(2,l)$-kernel for any $l \geq 2$.

Remark 2.4. The inequality announced in Theorem 2.3 can be strict for $l \geq 2$. Consider any k, $k > l+1$ and T_{k-1} has no (k,l)-kernel but that $L(D) \cong T_{k-2}$ has a kernel, and hence a (k,l)-kernel, for any $l \geq 2$.

Remark 2.5. As a direct consequence of Theorems 2.2 and 2.3 we obtain Theorem 1.1 in the case that each vertex has indegree at least one, as a kernel is a $(2,1)$-kernel. In addition, Theorem 1.2 is a direct consequence of Theorem 2.3, as a quasikernel is a $(2,2)$-kernel.

COROLLARY 2.1. *If D is a digraph such that each vertex has indegree at least one, then the number of $(2,l)$-kernels in D is less than or equal to the number of $(2,l)$-kernels in $L(D)$.*

The proof is a direct consequence of Theorem 2.3.

THEOREM 2.4. *Let D be a digraph such that every vertex has indegree at least one. If $L(D)$ has a (k,l)-kernel, then D has a (k',l')-kernel, for $k'+l \leq k$ and $l \leq l'$.*

Proof. Let D be a digraph as in the hypothesis, $\bar{K}$ a (k,l)-kernel of $L(D)$, $k'+l \leq k$, and $l \leq l'$. We will prove that $\bar{f}(\bar{K})$ is a (k',l')-kernel of D.

(a) If $\{x,y\} \subseteq \bar{f}(\bar{K})$, then $d_D(x,y) \geq k'$.

By contradiction, suppose that $d_D(x,y) = n < k'$, and let $(x = x_0, x_1, \ldots, x_n = y)$ be a shortest directed path from x to y in D. Since $x \in \bar{f}(\bar{K})$, there exists an arc $a = (u,x) \in \bar{K}$. Denoting $a_i = (x_{i-1}, x_i) \in A(D)$, $1 \leq i \leq n$, we have from Definition 1.1 that $(a, a_1, \ldots, a_n)$ is a directed path in $L(D)$ of length n. Now consider two possible cases.

If $a_n \in \bar{K}$, then $d_{L(D)}(a, a_n) \leq n < k' < k$ with $\{a, a_n\} \subseteq \bar{K}$, contradicting part (1) of Definition 1.6, as $\bar{K}$ is a (k,l)-kernel of $L(D)$.

If $a_n \notin \bar{K}$, then it follows from part (2) of Definition 1.6 that there exists $b \in \bar{K}$ such that $d_{L(D)}(a_n, b) \leq l$; let $(a_n = b_0, b_1, \ldots, b_m = b)$ be a shortest directed path in $L(D)$ from a_n to b. On the other hand, since $y = x_n \in \bar{f}(\bar{K})$, there exists $c = (v,y) \in \bar{K}$. Now consider two possibilities.

If $c \neq b$, then it follows from Definition 1.1 that $(c, b_1, b_2, \ldots, b_m = b)$ is a directed path in $L(D)$ from c to b in $L(D)$ of length $m \leq l < k$ with $\{c, b\} \subseteq \bar{K}$, contradicting part (1) of Definition 1.6, as $\bar{K}$ is a (k,l)-kernel of $L(D)$.

If $c = b$, then $(a, a_1, a_2, \ldots, a_n = b_0, b_1, \ldots, b_m = b)$ is a directed walk from a to b in $L(D)$ of length $n + m$; hence there exists in $L(D)$ a directed path from a to b of length at most $n + m$ and $n + m < k' + l \leq k$. So $d_{L(D)}(a, b) < k$, $a \neq n$ (because $x \neq y$, $a = (u, x)$,$b = c = (v, y)$), and $\{a, b\} \subseteq \bar{K}$. This contradicts part (1) of Definition 1.6, as $\bar{K}$ is a (k, l)-kernel of $L(D)$.

(*b*) If $x \notin \bar{f}(\bar{K})$, then there exists $y \in \bar{f}(\bar{K})$ such that $d_D(x, y) \leq l'$.

Let $x \in V(D) \setminus \bar{f}(\bar{K})$. It follows from the hypothesis of Theorem 2.4 that there exists $a = (u, x) \in A(D)$, and Definition 2.1 implies $a \notin \bar{K}$. Since $a \notin \bar{K}$ and $\bar{K}$ is a (k, l)-kernel of $L(D)$, it follows from part (2) of Definition 1.6 that there exists $b \in \bar{K}$ such that $d_{L(D)}(a, b) \leq l$; let $b = (v, y)$. Clearly $y \in \bar{f}(\bar{K})$ and $d_D(x, y) \leq l \leq l'$. □

THEOREM 2.5. *Let D be a digraph such that each vertex has indegree at least one. If $L(D)$ has a (k, l)-kernel $\bar{A}$ with the properties that $l < k$ and, for each arc $a \in \bar{A}$, there is an arc $b \neq a$ in $\bar{A}$ such that the terminal endpoints of a and b are the same, then $\bar{f}(\bar{A})$ is a (k, l)-kernel of D.*

Proof. Let D be a digraph and $\bar{A}$ a (k, l)-kernel of $L(D)$ as in the hypothesis of Theorem 2.5. We will prove that $\bar{f}(\bar{A})$ is a (k, l)-kernel of D.

(*a*) If $\{x, y\} \subseteq \bar{f}(\bar{A})$, $x \neq y$, then $d_D(x, y) \geq k$.

By contradiction, suppose that $d_D(x, y) = n < k$ and let $(x = x_0, x_1, \ldots, x_n = y)$ be a shortest directed path from x to y in D. Since $x \in \bar{f}(\bar{A})$, there exists $a = (u, x) \in \bar{A}$. Denote by $a_i = (x_{i-1}, x_i), 1 \leq i \leq n$ and consider the following two possible cases.

If $a_n = (x_{n-1}, y) \in \bar{A}$, then $(a, a_1, a_2, \ldots, a_n)$ is a directed path of length $n < k$ contained in $L(D)$ from a to a_n with $a \neq a_n$ and $\{a, a_n\} \subseteq \bar{A}$. This contradicts part (1) of Definition 1.6, as $\bar{A}$ is a (k, l)-kernel of $L(D)$.

If $a_n = (x_{n-1}, y) \notin \bar{A}$, it follows from part (2) of Definition 1.6 that there exists $b \in \bar{A}$ such that $d_{L(D)}(a_n, b) \leq l < k$. On the other hand, since $y \in \bar{f}(\bar{A})$, there exists $c = (v, y) \in \bar{A}$. Now consider two possibilities.

If $b \neq c$, we consider a shortest directed path from a_n to b, say $(a_n = b_0, b_1, \ldots, b_n = b)$, contained in $L(D)$; then it follows from Definition 2.1 that $(c, b_1, b_2, \ldots, b_n = b)$ is also a directed path in $L(D)$ of length $n < k$ from c to b with $c \neq b$ and $\{c, b\} \subseteq \bar{A}$, contradicting part (1) of Definition 1.6, as $\bar{A}$ is a (k, l)-kernel of $L(D)$.

If $b = c$, we consider an arc $d \in \bar{A}, d \neq b$ such that d and b have the same terminal endpoint (this is from the hypothesis of Theorem 2.5). It follows from Definition 2.1 that $(d, b_1, b_2, \ldots, b_n = b)$ is a directed path contained in $L(D)$ from d to b of length $n < k$ with $d \neq b$, $\{d, b\} \subseteq \bar{A}$, contradicting part (1) of Definition 1.6.

(*b*) If $x \notin \bar{f}(\bar{A})$, then there exists $y \in \bar{f}(\bar{A})$ such that $d_D(x, y) \leq l$.

It follows from the hypothesis of Theorem 2.5 that there exists an arc $a = (u, x) \in A(D)$; since $x \notin \bar{f}(\bar{A})$, we have $a \notin \bar{A}$. Now $a \notin \bar{A}$ and $\bar{A}$ is a (k, l)-kernel of $L(D)$, so there exists $b \in \bar{A}$ such that $d_{L(D)}(a, b) \leq l$. Let $(a = a_0, a_1, \ldots, a_n = b)$ be a shortest directed path in $L(D)$ from a to b, and $a_i = (x_{i-1}, x_i)$ for $1 \leq i \leq n$, $b = (x_{n-1}, x_n)$; then $(x, x_1, \ldots, x_{n-1}, x_n)$ is a directed walk in D of length $n \leq l$ from x to x_n; clearly Definition 2.1 implies $x_n \in \bar{f}(\bar{A})$. So, taking $y = x_n$, (*b*) is thus proved. □

COROLLARY 2.2. *Let D be a digraph such that each vertex has indegree at least one and let $1 \leq l < k$. If each (k, l)-kernel $\bar{A}$ of $L(D)$ satisfies that, for each arc $a \in \bar{A}$, there is an arc $b \in \bar{A}$ such that the terminal endpoints of a and b are the same, then the number of (k, l)-kernels of $L(D)$ is less than or equal to the number of (k, l)-kernels of D.*

Proof. Let $1 \leq l < k$, $\mathcal{K}_1$ be the set of all (k, l)-kernels of $L(D)$, let $\mathcal{K}$ be the set of all (k, l)-kernels of D, and let $\bar{f}' : \mathcal{K}_1 \to \mathcal{K}$ be the restriction of $\bar{f}$ to $\mathcal{K}_1$. From Theorem 2.5 it suffices to prove that $\bar{f}'$ is an injective function.

(*c*) If $\bar{K}_1 \neq \bar{K}_2$ and $\{\bar{K}_1, \bar{K}_2\} \subseteq \mathcal{K}_1$, then $\bar{f}'(\bar{K}_1) \neq \bar{f}'(\bar{K}_2)$.

Since $\bar{K}_1 \neq \bar{K}_2$, we can assume, without loss of generality, that $\bar{K}_1 \setminus \bar{K}_2 \neq \emptyset$. Let $a = (u,x) \in \bar{K}_1 \setminus \bar{K}_2$. It follows from Definition 2.1 that $x \in \bar{f}'(\bar{K}_1)$, and we will show that $x \notin \bar{f}'(\bar{K}_2)$.

By contradiction, suppose that $x \in \bar{f}'(\bar{K}_2)$. Hence there exists $b = (v,x) \in \bar{K}_2$. Since $a = (u,x) \notin \bar{K}_2$ and $\bar{K}_2$ is a (k,l)-kernel of $L(D)$, there exists $c \in \bar{K}_2$ such that $d_{L(D)}(a,c) \leq l < k$. Let $(a = a_0, a_1, \ldots, a_n = c)$ be a shortest directed path in $L(D)$ from a to c and consider the following two possibilities:

If $b \neq c$, then it follows from Definition 2.1 that $(b, a_1, a_2, \ldots, a_n = c)$ is a directed path in $L(D)$ from b to c of length $n \leq l < k$ with $\{b,c\} \subseteq \bar{K}_2$, $b \neq c$. This contradicts part (1) of Definition 1.6, as $\bar{K}_2$ is a (k,l)-kernel of $L(D)$.

If $b = c$, we have from the hypothesis of Corollary 2.2 that there exists an arc $d \in \bar{K}_2$ such that $d \neq b$ and that d and b have the same terminal endpoint x. Then it follows from Definition 2.1 that $(d, a_1, a_2, \ldots, a_n = c = b)$ is a directed path in $L(D)$ of length $n \leq k < l$ from d to b with $d \neq b$, $\{d,b\} \subseteq \bar{K}_2$, contradicting part (1) of Definition 1.6, as $\bar{K}_2$ is a (k,l)-kernel of $L(D)$. □

THEOREM 2.6. *Every digraph has a $(k, 2k-2)$-kernel.*

Proof. We proceed by induction on $|V(D)|$.

For D with $|V(D)| = 1$ it is obvious. Suppose that if D' is a digraph with $|V(D')| < n$, then D' has a $(k, 2k-2)$-kernel, and let D be a digraph with $|V(D)| = n$.

Let $x_0 \in V(D)$ and $D^* = D[V(D) \setminus \{x \in V(D) | d_D(x, x_0) \leq k-1\}]$. Clearly $|V(D^*)| < n$, and hence D^* has a $(k, 2k-2)$-kernel, namely S^*. Consider the following two possibilities.

If there exists a directed path in D of length less than or equal to $k-1$, then S^* is a $(k, 2k-2)$-kernel of D.

If there is no directed path in D from x_0 to some point of S^* of length less than or equal to $k-1$, then $S^* \cup \{x_0\}$ is a $(k, 2k-2)$-kernel of D. □

COROLLARY 2.3. *Every digraph has a (k,l)-kernel for $l \geq 2k-2$.*

The proof is a direct consequence of Theorem 2.6 and Definition 1.6, as a (k,l)-kernel of a digraph D is also a (k,l')-kernel for every $l' \geq l$.

Remark 2.6. The hypothesis $l \geq 2k-2$ cannot be omitted in Corollary 2.3. Consider C_{2k-1} to be the directed cycle of length $2k-1$; for any $l < 2k-1$, the digraph C_{2k-1} has no (k,l)-kernel.

COROLLARY 2.4 (see [3]). *Every digraph has a quasikernel.*

The proof is a direct consequence of Theorem 2.6 by taking $k = 2$, as a quasikernel is a $(2,2)$-kernel.

REFERENCES

[1] C. BERGE, *Graphs*, North-Holland, Amsterdam, New York, 1985.

[2] C. BERGE AND A. RAMACHANDRA RAO, *A combinatorial problem in logic*, Discrete Math., 17 (1977), pp. 23–26.

[3] V. CHVÁTAL AND L. LOVÁSZ, *Every directed graph has a semi-kernel*, in Hypergraph Seminar, Lecture Notes in Math. 441, Springer-Verlag, Berlin, 1974, p. 175.

[4] P. DUCHET, *A sufficient condition for a digraph to be kernel-perfect*, J. Graph Theory, 11 (1987), pp. 81–85.

[5] P. DUCHET AND H. MEYNIEL, *Une généralization du théoréme de Richardson sur l'existence de noyaux dans les graphes orientés*, Discrete Math., 43 (1983), pp. 21–27.

[6] P. DUCHET, Y.O. HAMIDOUNE, AND H. MEYNIEL, *Sur les quasi-noyaux d'un graphe*, Discrete Math., 65 (1987), pp. 231–235.

[7] H. GALEANA-SÁNCHEZ AND V. NEUMANN-LARA, *On kernels and semikernels of digraphs*, Discrete Math., 48 (1984), pp. 67–76.

[8] H. Galeana-Sánchez, L. Pastrana-Ramirez, and H.A. Rincón-Mejia, *Semikernels, quasikernels, and grundy functions in the line digraph*, SIAM J. Discrete Math., 4 (1991), pp. 80–83.
[9] M. Harminc, *Solutions and kernels of a directed graph*, Math. Slovaca, 32 (1982), pp. 263–267.
[10] M. Kwaśnik, *The generalization of Richardson theorem*, Discussiones Math., IV (1981), pp. 11–14.
[11] J. Von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1944.
[12] V. Neumann-Lara, *Seminúcleos en una digráfica*, Anales del Instituto de Matemáticas de la Universidad Nacional Autónoma de México, 11 (1971), pp. 55–62.
[13] M. Richardson, *Solution of irreflective relations*, Ann. Math., 58 (1953), pp. 573–580.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 347–372, August 1998

MATCHING NUTS AND BOLTS IN $O(n \log n)$ TIME*

JÁNOS KOMLÓS†, YUAN MA‡, AND ENDRE SZEMERÉDI§

Abstract. Given a set of n nuts of distinct widths and a set of n bolts such that each nut corresponds to a unique bolt of the same width, how should we match every nut with its corresponding bolt by comparing nuts with bolts? (No comparison is allowed between two nuts or two bolts.) The problem can be naturally viewed as a variant of the classic sorting problem as follows. Given two lists of n numbers each such that one list is a permutation of the other, how should we sort the lists by comparisons only between numbers in different lists? We give an $O(n \log n)$-time deterministic algorithm for the problem. This is optimal up to a constant factor and answers an open question posed by Alon et al. [*Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994, pp. 690–696]. Moreover, when copies of nuts and bolts are allowed, our algorithm runs in optimal $O(\log n)$ time on n processors in Valiant's parallel comparison tree model. Our algorithm is based on the AKS sorting algorithm with substantial modifications.

Key words. sorting, matching, selection, parallel computation, AKS sorting algorithm, random graphs

AMS subject classifications. 68Q20, 68Q22, 68Q25, 68R10, 06A05

PII. S0895480196304982

1. Introduction.

Given a set of n nuts of distinct widths and a set of n bolts such that each nut corresponds to a unique bolt of the same width, how should we match every nut with its corresponding bolt by comparing nuts with bolts? (No comparison is allowed between two nuts or two bolts.)

This problem can be naturally viewed as a variant of the classic sorting problem as follows. Given two lists of n numbers each such that one list is a permutation of the other, how should we sort the lists by comparisons only between numbers in different lists? In fact, the following simple reasoning illustrates that the problem of matching nuts and bolts and the problem of sorting them have the same complexity, up to a constant factor. On one hand, if the nuts and bolts are sorted, then a nut and a bolt at the same position in the sorted order certainly match each other. On the other hand, if the nuts and bolts are matched, we can sort them by any optimal sorting algorithm in $O(n \log n)$ time. Hence, the complexity equivalence of sorting

*Received by the editors June 7, 1996; accepted for publication (in revised form) June 26, 1997; published electronically July 7, 1998. A preliminary version of this paper appeared in *Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, Atlanta, GA, 1995, pp. 232–241. This work is part of the "Hypercomputing and Design" (HPCD) project, and it is supported in part by ARPA under contract DABT-63-93-C-0064. The content of the information herein does not necessarily reflect the position of the government and official endorsement should not be inferred.

http://www.siam.org/journals/sidma/11-3/30498.html

†Department of Mathematics, Rutgers University, Piscataway, NJ 08855 (komlos@math.rutgers.edu).

‡Department of Computer Science, Stanford University, Stanford, CA 94305. Present address: Haas School of Business, University of California at Berkeley, Berkeley, CA 94720 (yuan@haas.berkeley.edu). The research of this author was supported in part by an NSF Mathematical Sciences Postdoctoral Research Fellowship. Part of the work was done while the author was visiting DIMACS, and part of the work was done while at MIT and supported by DARPA contracts N00014-91-J-1698 and N00014-92-J-1799.

§Department of Computer Science, Rutgers University, Piscataway, NJ 08855 (szemered@cs.rutgers.edu). Part of the work was done while the author was at the University of Paderborn, Paderborn, Germany.

and matching them follows from the simple information lower bound of $\Omega(n \log n)$ on the matching problem, which can be easily derived from the fact that there are $n!$ possible ways to match the nuts and bolts. So in this paper we will consider the problem of how to sort the nuts and bolts instead of the problem of matching them.

The problem of sorting nuts and bolts has a simple randomized algorithm (e.g., a simple variant of the QUICKSORT algorithm) that runs in the optimal $O(n \log n)$ expected time [10]. However, finding a nontrivial (say, $o(n^2)$-time) deterministic algorithm has appeared to be highly nontrivial. Alon et al. [3] designed an $O(n \log^4 n)$-time deterministic algorithm based on expander graphs, and they posed the open question of designing an optimal deterministic algorithm to the problem. Recently, Bradford and Fleischer [8] improved the running time to $O(n \log^2 n)$, but the question remains open if $O(n \log n)$ can be achieved.

Since the classic sorting problem has been intensively studied, it is natural to ask if any existing $O(n \log n)$-time deterministic sorting algorithm can be easily adapted to sort nuts and bolts. In a certain sense, most of the existing $O(n \log n)$-time sorting algorithms use a divide-and-conquer approach. In particular, they require recursive solutions to subproblems of smaller sizes. For the classic sorting problem, solving the subproblems is simple. However, in the context of sorting nuts and bolts, solving a subproblem can raise many problems. In particular, the fact that we can sort the nuts and bolts at all relies on the fact that there is a match between them.[1] For example, if all of the nuts happen to be smaller than all of the bolts, then we will not be able to learn anything about the order of the nuts or the order of the bolts by comparing nuts against bolts only. As a consequence, if we want to make use of existing sorting algorithms, it is essential to make arrangements so that, when we work on a smaller set of nuts and a smaller set of bolts, we may obtain useful information in an efficient way. Unfortunately, no existing deterministic sorting algorithm of $O(n \log n)$ time seems readily adaptable to make such arrangements.

Faced with such difficulty, the algorithm of Alon et al. [3] uses an $O(n \log^3 n)$-time algorithm for selecting a median nut and a median bolt, which, in turn, is based on expander graphs. However, as pointed out by Alon et al. [3], this particular method cannot be adapted to select a median in $O(n)$ time, and a possible $O(n \log n)$ algorithm needs to come from a different means. Similarly, the $O(n \log^2 n)$-time algorithm of Bradford and Fleischer [8] is based on an $O(n \log n)$-time algorithm for selecting a median nut and a median bolt. In fact, we have discovered a (fairly) simple $O(n(\log \log n)^2)$-time algorithm for selecting a median nut and a median bolt, thereby giving an $O(n \log n \, (\log \log n)^2)$-time algorithm for sorting nuts and bolts. We will not give any details of this algorithm, however, since it appears that we need to do something very different to achieve an optimal $O(n \log n)$ time.

The main contribution of this paper is an $O(n \log n)$-time algorithm for sorting nuts and bolts, which is based on the AKS sorting algorithm [2] with substantial modifications.[2] As a by-product of our AKS-based approach, our algorithm can be executed in $O(\log n)$ time on n processors in Valiant's parallel comparison tree model [11] when copying of nuts and bolts is allowed. In Valiant's model, only com-

[1] Such a condition can be slightly relaxed, as to be discussed in section 4.

[2] The AKS sorting algorithm was designed to be implemented in an oblivious fashion on a *comparator network*, and it also has an optimal parallel running time of $O(\log n)$ on n processors [2]. In this paper, our main focus is the sequential algorithm model, and we will refer to the work of [2] as the AKS sorting *algorithm*, as opposed to the AKS sorting *network*.

parisons are counted toward the running time, and bookkeeping is free. We remark that our algorithm is not fully constructive, and some of its gadgets depend on some random graph properties. The existence of such graphs is easily proven by a random construction, but we do not know how to construct them explicitly. However, all other parts of our algorithm are constructive, and once explicit constructions of the desired graphs are discovered, our algorithm will be constructive as well.

The rationale of using an AKS-based approach for sorting nuts and bolts lies behind some special properties of the AKS sorting algorithm. Roughly, as described by Paterson [9], the AKS sorting algorithm proceeds as follows. It arranges the numbers being sorted in a complete binary tree, which will be referred to as the AKS tree. Each node of the AKS tree contains a set of numbers. Most of the numbers in the same node have ranks within a certain interval. At each stage of the algorithm, a certain sorting-related device (with $O(1)$ parallel time) is used to approximately partition the numbers at each node of the AKS tree. In a way, the AKS sorting algorithm proceeds by partitioning in a weak sense: it *approximately* partitions numbers into *almost* correct halves and has an intricate error-correcting mechanism. In particular, unlike most other known $O(n \log n)$-time deterministic sorting algorithms, the AKS sorting algorithm does not proceed in a rigorous divide-and-conquer fashion. These special properties will appear to be advantageous in sorting nuts and bolts.

Although there are good reasons why the AKS sorting algorithm may be a good tool for sorting nuts and bolts, a direct modification of the AKS sorting algorithm does not solve our problem. For example, one naive approach is as follows. Keep two AKS trees, one for the nuts and the other for the bolts; at each stage of the algorithm, compare nuts and bolts in corresponding AKS tree nodes according to an expander graph, and reallocate the nuts and bolts according to the results of the comparisons. Such an approach proceeds well at a few initial stages, but it has serious troubles in future stages. The problem arises since we cannot keep a match between the nuts and bolts in corresponding AKS tree nodes. For example, when the roots contain only a constant number of nuts and bolts, it is possible that all of the nuts contained in the root of one AKS tree are smaller than all of the bolts contained in the root of the other AKS tree, in which case we cannot obtain any information, by comparisons between the nuts and bolts in the roots, about the order of the nuts or the order of the bolts that are located in the roots. In fact, such observations may even lead one to question whether the AKS-based approach is helpful at all for sorting nuts and bolts. The novelty of our work in adapting the AKS sorting algorithm is in introducing certain mechanisms that allow efficient approximate-partitioning at an AKS tree node even if the nuts and bolts in the corresponding AKS tree nodes do not form a match.

The remainder of the paper is organized into sections as follows. In section 2, we present our algorithm for sorting nuts and bolts. In section 3, we prove the correctness of the algorithm and analyze its running time. We conclude in section 4 with discussions on some extensions and open problems.

2. An $O(n \log n)$-time algorithm for sorting nuts and bolts. This section contains the description of our $O(n \log n)$-time algorithm for sorting nuts and bolts. As pointed out in the introduction, our algorithm depends on some random graphs, which we do not know how to construct explicitly. Also, we will be content with an algorithm of $O(n \log n)$ running time and will not attempt to keep the involved

constants small. In particular, a large constant (much larger than the best previously known constant for the AKS sorting algorithm) is hidden behind the "O" notation.

2.1. An overview of the algorithm. In this subsection, we give a high-level description of our AKS-based algorithm. This algorithm proceeds much like the AKS sorting algorithm, except that we use a completely different method to partition nuts (bolts) in an AKS tree node. The partition method is fairly complicated and will be the subject of the next subsection. In this subsection, we will assume such a partition can be done and focus on other simpler issues. In the rest of the paper, we will assume without loss of generality that n is an integer power of 2, since otherwise we may include some dummy nuts and bolts that are larger than all of the nuts and bolts in the original problem.

We first need a complete understanding of the AKS sorting algorithm. However, since the AKS sorting algorithm is fairly complicated, we will only sketch the AKS sorting algorithm at a high level, and we refer the readers to [9] for a complete and rigorous description of the AKS sorting algorithm.

As described in [9], the AKS sorting algorithm arranges all the numbers being sorted within a complete binary tree, with the root at the top. A rigorous treatment of the tree structure can be found in [9]. We will refer to such a tree as an *AKS tree* and refer to a node in the tree as an *AKS tree node*. The numbers being sorted are located within the AKS tree nodes. Each AKS tree node X has a capacity, denoted by $\mathrm{cap}(X)$, that specifies the maximum number of numbers that can be contained in X. Let $|X|$ denote the number of numbers that are indeed contained in X; X is called empty, full, or partially full if $|X| = 0$, $|X| = \mathrm{cap}(X)$, or $0 < |X| < \mathrm{cap}(X)$, respectively. The AKS sorting algorithm works in stages, starting from stage one. Within each stage there is a sorting-related device that partitions each AKS tree node, X, into four parts, FL, CL, CR, and FR, which stand for "far-left," "center-left," "center-right," and "far-right," respectively. (To be rigorous, we partition the *list of numbers* in X, as opposed to X itself. But we will not distinguish X from the list of numbers contained in X when no confusion can arise.) By doing so, we hope to move most of the numbers in X into the correct halves, $FL \cup CL$ and $CR \cup FR$, and to move most of the "extreme" numbers to the extreme positions in FL and FR. At the *end* of a stage, numbers in FL and FR are sent to the parent of X, and CL and CR are sent to the left and right children of X, respectively. This will have the effect of moving most of the correctly located numbers downward in the AKS tree and moving most of the incorrectly located numbers upward in the AKS tree. Overall, most numbers in the lower part of the AKS tree are near their correct positions, and numbers far away from their correct positions tend to move upward in the AKS tree so that they will be processed further. The AKS tree can be viewed to be infinite, but we make the convention that a *leaf* of an AKS tree is a lowest nonempty AKS tree node. At odd stages, all nodes at odd levels and all nodes below the leaf level are empty, and all nodes at even levels above the leaf level are full except that nodes at the leaf level can be full or partially full.[3] (The root is assumed to be at level *0*.) The opposite holds at even stages. This completes our brief description of the AKS sorting algorithm.

At a high level, our algorithm differs from the original AKS sorting algorithm in

[3]To be more rigorous, when we say that a node is full or empty during a stage, we mean it is full or empty at the beginning of the stage. Note that numbers in a full or partially full node X will be moved to the parent or children of X at the end of the stage.

two ways: (1) we need to keep two separate AKS trees: (T_N for the set of nuts N, and T_B for the set of bolts B); (2) we need a completely different method to partition *elements* (which refer to nuts or bolts) in an AKS tree node.

Other than these two differences, our algorithm for sorting nuts and bolts works exactly as the AKS sorting algorithm does. In particular, the structures of the two AKS trees are identical (except one contains nuts and the other contains bolts): T_N and T_B are each specified by the same set of parameters as in the AKS sorting algorithm. To describe explicitly how our algorithm works, we need to specify some parameters associated with the AKS sorting algorithm. For simplicity, we will explicitly follow the parameter choices of [9] whenever possible. In particular, we will use the same letters to denote the same quantities as in [9] unless specified otherwise.

We choose the same parameters associated with our AKS trees as in [9]:

$$A = 3,\ \nu = \frac{43}{48},\ \text{and}\ \lambda = \frac{1}{8}.$$

As in [9], the choices of these parameters completely determine how the nuts and bolts move within T_N and T_B. In particular,

- the capacity of an AKS tree node X immediately after stage t at level d is determined by $\text{cap}(X) = \nu^t A^d N(1 - \frac{1}{4A^2})$;
- at each stage, the elements at X are partitioned into four parts, FL, CL, CR, and FR, such that (1) $|FL| = |FR| = \min\{\frac{\lambda}{2}\,\text{cap}(X), \frac{|X|}{2}\}$, $|CL| = |CR| = \frac{|X|}{2} - |FL|$ and (2) at the end of the stage, FL and FR are moved to the parent of X, and CL and CR are moved to the left and right children of X, respectively.

Also, we choose

$$\mu = \frac{1}{36}\ \text{and}\ \delta = \frac{1}{40},$$

the same as in [9]. Note that μ and δ have nothing to do with the description of the algorithm and will be used only in the analysis of the algorithm.

Another parameter ε was used in [9] to specify the functionality of the so-called *separator*, which corresponds to the so-called *near-sorting network* of [2]. In [9], a separator is used to partition an AKS tree node X into four parts: FL, CL, CR, and FR. In our algorithm, however, we cannot use a separator or near-sorting network since, as we have explained in the introduction, we cannot enforce a match between nuts and bolts in corresponding AKS tree nodes. Nevertheless, we need a sorting-related device for such a partition. The partition scheme is fairly intricate and will be the subject of the next subsection. In any event, following the notation of [9], we will use parameter ε to measure the accuracy of our partition method. We do not specify how to choose ε explicitly. Instead, we will be content with proving that a sufficiently small ε suffices for our purposes.

Finally, as in [9], we also need to deal with the so-called boundary conditions and integer rounding. These can be easily handled in the same way as in [9], and we will not address these particular technical problems hereafter.

2.2. Partitioning nuts or bolts at an AKS tree node. In this subsection, we describe an algorithm to partition elements in an AKS tree node X into four parts: FL, CL, CR, and FR. We will accomplish the partition of X by comparing

nuts (or bolts) in X with bolts (or nuts) in a set $S(X)$, which is to be defined in subsection 2.2.2. On one hand, $S(X)$ should be large enough so that a proper partition of X is possible; i.e., $S(X)$ should contain enough bolts (or nuts) to separate some of the nuts (or bolts) in X from the others. On the other hand, $S(X)$ should be small enough so that the number of necessary comparisons between X and $S(X)$ for partitioning X is not prohibitively large.

The remainder of the subsection is organized as follows. In subsection 2.2.1, we prove a lemma on random graphs and describe how to use the graphs to construct a comparison algorithm. In subsection 2.2.2, we construct $S(X)$. In subsection 2.2.3, we describe how to partition X by applying the comparison algorithm of subsection 2.2.1 to X and $S(X)$.

2.2.1. Random graphs and a comparison algorithm. In this subsection, we first prove a useful lemma on random bipartite graphs. Then, we describe how to use such graphs in a comparison algorithm, which is an important building block in our $O(n \log n)$-time algorithm for sorting nuts and bolts. Although a random graph will yield a desired graph with high probability, we do not know how to construct such graphs explicitly.

The graphs considered in this paper are allowed to be multigraphs, and we use $e(X, Y)$ to denote the number of edges between X and Y for arbitrary vertex subsets X and Y. In particular, if there are m edges between a vertex $u \in X$ and a vertex $v \in Y$, then each of the m multiple edges between u and v is counted exactly once in $e(X, Y)$. Also, we use "e" to denote the natural number and "ln" to denote the logarithm with base e. We remark that the parameter ϵ in the next lemma should be distinguished from ε, which is alluded to near the end of subsection 2.1 and will be made more explicit in Property 3.2.

LEMMA 2.1. *Let ϵ and θ be two arbitrary constants in $(0, 1)$, and let U and V be two sets such that $|U| \le |V|$. If $d \ge 2\epsilon^{-3} \ln\big((e^2/\epsilon^2)(|V|/|U|)\big)$, then there exists a bipartite graph $G = (U, V, E)$, $E \subseteq U \times V$ with the following properties:* (1) *$deg(v) = d$ for all $v \in V$;* (2) *$e(X, Y) \ge (1 - \epsilon)d\,|X|\,|Y|/|U|$ for any sets $X \subseteq U$, $Y \subseteq V$ such that $|X| \ge \epsilon|U|$ and $|Y| \ge \epsilon|U|$; and* (3) *if $|U| = |V|$ and $d \ge (8/\theta)\ln(16/\theta)$, then any $Y \subseteq V$ of size $|Y| \le 2\,e^{-4}\theta\,|U|/d$ is directly connected (i.e., connected by an edge, as opposed to by a path) to at least $\theta d|Y|/2$ vertices in U, even after an arbitrary set of up to $(1 - \theta)d$ edges are removed from each vertex in Y.*

Proof. We prove the lemma by giving a random construction that yields a desired graph with high probability. We construct our random graph $G \subseteq U \times V$ in d rounds. In each of the d rounds, for every $v \in V$, choose a vertex $u \in U$ uniformly at random and include edge (u, v) in E. That is, each u in U is chosen with probability $\frac{1}{|U|}$, independently of the choices of vertices of any previous rounds and the choices of vertices within the same round for other vertices in V. Thus, an edge between a fixed pair of vertices u and v may be selected more than once (i.e., G may be a multigraph), and the degree of a vertex in U can be anything between 0 and $d|V|$. However, the degree of each vertex in V is exactly d, meaning that G satisfies property (1) of the lemma.

We next need to show that G satisfies properties (2) and (3) with nonzero probability. Thus, it suffices to show

$$\Pr\left(G \text{ does not satisfy property (2)}\right) \le \frac{1}{2\pi} \tag{1}$$

and

$$\Pr\left(G \text{ does not satisfy property (3)}\right) \le \frac{1}{2\pi}. \tag{2}$$

We first prove inequality (1). Consider an arbitrary pair of vertex subsets $X \subseteq U$ and $Y \subseteq V$ such that $|X| \ge \epsilon|U|$ and $|Y| \ge \epsilon|U|$. According to our construction, there are exactly $d|Y|$ edges associated with nodes in Y. Each of these edges has its other node in X with probability $\frac{|X|}{|U|}$. Hence, by the Chernoff bound [4, Theorem A.13, p. 238],

$$\Pr\left(e(X,Y) < (1-\epsilon)\, d\, \frac{|X|}{|U|}\, |Y| \right) \le e^{-\frac{\epsilon^2 d\, |X|\, |Y|}{2\,|U|}} \le e^{-\frac{\epsilon^3 d\, |Y|}{2}}, \tag{3}$$

where the last inequality follows from the assumption $|X| \ge \epsilon|U|$. On the other hand, it is easy to see that if there are two disjoint vertex sets A, B with "edge density" $e(A,B)/(|A||B|)$ between them and if a, b are integers such that $a \le |A|$ and $b \le |B|$, then there are two sets $X \subseteq A$ and $Y \subseteq B$ such that $|X| = a$, $|Y| = b$, and $e(X,Y)/(|X||Y|) \le e(A,B)/(|A||B|)$. Thus, writing $\ell = \lceil \epsilon|U| \rceil$, we have

$$\begin{aligned}
&\Pr\Bigg(\exists\, X, Y \text{ subject to (s.t.) } X \subseteq U,\ Y \subseteq V,\ |X| \ge \epsilon|U|,\ |Y| \ge \epsilon|U|, \\
&\qquad \text{and } e(X,Y) < (1-\epsilon)\, d\, \frac{|X|}{|U|}\, |Y| \Bigg) \\
&\le \Pr\Bigg(\exists\, X, Y \text{ s.t. } X \subseteq U,\ Y \subseteq V,\ |X| = \lceil \epsilon|U| \rceil,\ |Y| = \lceil \epsilon|U| \rceil, \\
&\qquad \text{and } e(X,Y) < (1-\epsilon)\, d\, \frac{|X|}{|U|}\, |Y| \Bigg) \\
&\le \binom{|U|}{\ell} \binom{|V|}{\ell} e^{-\frac{\epsilon^3 d\, \ell}{2}} \\
&\le \frac{1}{2\pi} \left(\frac{e|U|}{\ell}\, \frac{e|V|}{\ell}\, e^{-\frac{\epsilon^3 d}{2}} \right)^{\ell},
\end{aligned}$$

where the first inequality is indeed an equality (although we do not need this fact), the second inequality follows from inequality (3), and the last inequality follows from the inequality

$$\binom{x}{y} \le \frac{1}{\sqrt{2\pi}} \left(\frac{ex}{y} \right)^{y} \text{ for } y \ge 1, \tag{4}$$

which may be verified by Stirling's formula. Now, inequality (1) follows since our assumption $d \ge 2\epsilon^{-3} \ln\left((e^2/\epsilon^2)(|V|/|U|)\right)$ implies

$$\frac{e|U|}{\ell}\, \frac{e|V|}{\ell}\, e^{-\epsilon^3 d/2} \le 1.$$

We next prove inequality (2). Let

$$y = \frac{2e^{-4}\theta}{d}. \tag{5}$$

If G does not satisfy property (3), then there exists a set $Y \subseteq V$ of size $|Y| = k \le y|U|$ such that after the removal of $\lfloor (1-\theta)d \rfloor$ edges from each vertex in Y, the set Y is only directly connected to vertices in a set $X \subseteq U$ with $|X| = \ell(k) = \lfloor \theta dk/2 \rfloor$. Writing $r = \lceil \theta d \rceil$, we know that the probability that such a set Y exists is at most

$$\sum_{k=1}^{y|U|} \binom{|V|}{k} \binom{|U|}{\ell(k)} \binom{d}{r}^k \left(\frac{\ell(k)}{|U|}\right)^{rk} \le (2\pi)^{-3/2} \sum_{k=1}^{y|U|} \left(\frac{e|V|}{k}\right)^k \left(\frac{e|U|}{\ell(k)}\right)^{\ell(k)} \left(\frac{ed\ell(k)}{r|U|}\right)^{rk}, \tag{6}$$

where the inequality follows from inequality (4). By equality (5) and the definition of k, it is easy to verify $\ell(k) \le \theta\, d\, k/2 \le |U|$ and $r \ge \theta d \ge d\ell(k)/|U|$. Thus, since the function $(eA/x)^x$ is increasing for $x \le A$ and decreasing for $x \ge A$, the right-hand side of inequality (6) is at most

$$\begin{aligned}
&(2\pi)^{-3/2} \sum_{k=1}^{y|U|} \left(\frac{e|V|}{k}\right)^k \left(\frac{e|U|}{\theta dk/2}\right)^{\theta dk/2} \left(\frac{ed\ell(k)}{\theta d|U|}\right)^{\theta dk} \\
&\le (2\pi)^{-3/2} \sum_{k=1}^{y|U|} \left[\left(\frac{e|V|}{k}\right) \left(\frac{e|U|}{\theta dk/2}\right)^{\theta d/2} \left(\frac{edk/2}{|U|}\right)^{\theta d} \right]^k \qquad \left(\text{since } \ell(k) \le \frac{\theta dk}{2}\right) \\
&= (2\pi)^{-3/2} \sum_{k=1}^{y|U|} \left[\left(\frac{e|V|}{k}\right) \left(\frac{e^4 dk}{2\theta|U|}\right)^{\theta d/2} e^{-\theta d/2} \right]^k.
\end{aligned}$$

It suffices to show that, for any k in the range of the summation, the expression in the last square bracket is at most $1/2$ (so that we can upper bound the sum by a geometric series). Since $\theta d/2 \ge 1$, this reduces to show

$$\frac{e^4 dk}{2\theta|U|} \le 1 \text{ and } \frac{e^5 d}{2\theta} \frac{|V|}{|U|} e^{-\theta d/2} \le \frac{1}{2}.$$

The first inequality follows from $k \le y|U|$ and equality (5). Since we assume $|U| = |V|$ and $d \ge \frac{8}{\theta} \ln \frac{16}{\theta}$ for property (3), the second inequality reduces to verify $f(d) = -\frac{e^5 d}{\theta} + e^{\frac{d\theta}{2}} \ge 0$ for $d \ge \frac{8}{\theta} \ln \frac{16}{\theta}$. This inequality holds since $f'(d) \ge 0$ for $d \ge \frac{8}{\theta} \ln \frac{16}{\theta}$ and

$$f\left(\frac{8}{\theta} \ln \frac{16}{\theta}\right) = \left(\frac{16}{\theta}\right)^4 - \frac{e^5}{\theta} \frac{8}{\theta} \ln \frac{16}{\theta} \ge \left(\frac{16}{\theta}\right) \left(\left(\frac{16}{\theta}\right)^3 - \frac{8\, e^5}{\theta^2} \right) \ge 0. \qquad \square$$

Roughly, Lemma 2.1 says that the number of edges between two sets of vertices cannot be much smaller than the average number of edges between two sets of their sizes. In a certain sense, this also means that the edges between U and V are evenly distributed, and so the number of edges between two sets of vertices cannot be much larger than the average. Formally, we have the following corollary.

COROLLARY 2.2. *In a graph that satisfies properties* (1) *and* (2) *of Lemma* 2.1, *for any sets* $X \subseteq U$ *and* $Y \subseteq V$ *such that* $|Y| \ge \epsilon|U|$, $e(X,Y) \le d\,|X||Y|/|U| + \epsilon\, d\,|Y|$.

Proof. If $|X| > (1-\epsilon)|U|$, then the right-hand side of the desired inequality is clearly larger than $d\,|Y|$, which, in turn, is greater than or equal to the left-hand side of the desired inequality. Hence, we will assume $|X| \le (1-\epsilon)|U|$. Applying the second

statement of Lemma 2.1 to the sets $U - X$ and Y, we obtain

$$\begin{aligned} e(U - X, Y) &\geq (1 - \epsilon)\, d\, |U - X|\, |Y|/|U| \\ &= (1 - \epsilon)\, d\, |Y| - (1 - \epsilon)\, d\, |X||Y|/|U|. \end{aligned}$$

On the other hand, the first statement of Lemma 2.1 implies $e(U, Y) = d\,|Y|$. Hence,

$$\begin{aligned} e(X, Y) &= d\,|Y| - e(U - X, Y) \\ &\leq \epsilon\, d\,|Y| + (1 - \epsilon)\, d\,|X||Y|/|U| \\ &\leq d\,|X||Y|/|U| + \epsilon\, d\,|Y|. \qquad \square \end{aligned}$$

We now describe how to apply the graph of Lemma 2.1 to construct a comparison algorithm in a way similar to that of [2] and [9]. We will use some adaptive methods, such as counting, in some future applications of the algorithm, whereas [2] and [9] deal with comparator networks and can only use oblivious methods. Given an arbitrary set of nuts (bolts) U, an arbitrary set of bolts (nuts) V with $|V| \geq |U|$, and a bipartite graph $G \subseteq U \times V$, Algorithm COMPARE(U, V, G) works as follows.

ALGORITHM COMPARE(U, V, G).

Step 1. Set SMALL(v) = LARGE$(v) = 0$ for each $v \in V$.

Step 2. For each edge (u, v) in graph G, compare u and v. Then, increment SMALL(v) by 1 if $v < u$; increment LARGE(v) by 1 if $v > u$; increment SMALL(v) and LARGE(v) each by $1/2$ if $v = u$.

In the above algorithm, SMALL(v) (resp., LARGE(v)) denotes the number of comparisons in Algorithm COMPARE where v is strictly smaller than (respectively, strictly larger than) its opponent plus half of the number of comparisons in Algorithm COMPARE where v is equal to (respectively, equal to) its opponent. In particular, we increment both LARGE(v) and SMALL(v) by $1/2$ if v is equal to its opponent. Such an arrangement will make some of our future arguments simple by ensuring that the values of SMALL and LARGE are symmetric. We remark that there may be multiple edges between u and v, in which case u and v are compared more than once and SMALL(v) or LARGE(v) is updated every time a comparison between u and v occurs.

It would be nice if Algorithm COMPARE(U, V, G) always provides an approximate partition of V. However, such a partition is not always possible. For example, if every nut in V is smaller than every bolt in U, then no matter how we conduct our comparisons, the outcome will not provide any useful information for partitioning V. Nevertheless, we next show that the algorithm has a certain ranking property in a certain case. Such a ranking property will then be further exploited to provide a more sophisticated algorithm for partition.

In what follows, we define the *rank* of an element x with respect to (with respect to) a set Y, denoted by $\text{rank}(x, Y)$, as the number of elements in Y that are smaller than or equal to x. Note that $\text{rank}(x, Y)$ is well defined even if x and elements of Y cannot be compared by a direct comparison; e.g., x and all elements in Y are nuts. When we say the rank of element x, denoted by $\text{rank}(x)$, without specifying a corresponding Y, we mean the rank of x with respect to B (or, equivalently, with respect to N). For any $\zeta, \xi \in [0, 1]$ and for any sets of elements U and V, let

$$V(\zeta, \xi, U) = \{v \in V \mid \zeta\,|U| \leq \text{rank}(v, U) \leq \xi\,|U|]\}.$$

LEMMA 2.3. *Assume U and V are a set of nuts and a set of bolts (or a set of bolts and a set of nuts), resp., $\epsilon|U| \geq 2$, $\zeta, \xi \in [0, 1]$, and $G \subseteq U \times V$ is a bipartite graph with*

parameters d and ϵ as described in Lemma 2.1. *(G is not required to satisfy the third property of Lemma* 2.1*, so the parameter θ of Lemma* 2.1 *is not described here.) If Algorithm* COMPARE(U, V, G) *is executed, then* (1) *at most $\epsilon|U|$ elements in $V(0, \xi, U)$ have their* SMALL *values less than or equal to $(1-\xi-2\epsilon)d$,* (2) *at most $\epsilon|U|$ elements in $V(\zeta, 1, U)$ have their* LARGE *values less than or equal to $(\zeta - 2\epsilon)d$, and* (3) *for any $X \subseteq V(0, \zeta, U)$ and any $Y \subseteq V(\xi, 1, U)$, where $\xi - \zeta \geq 6\,\epsilon$, if* SMALL$(x) \leq$ SMALL(y) *for all $x \in X$ and all $y \in Y$, then either $|X| < \epsilon|U|$ or $|Y| < \epsilon|U|$.*

Proof. We first prove the first claim in the lemma. Let

$$V_S = \{v \in V(0, \xi, U) \,|\, \text{SMALL}(v) \leq (1 - \xi - 2\,\epsilon)\,d\}.$$

We need to prove $|V_S| \leq \epsilon\,|U|$. We may assume without loss of generality

$$1 - \xi \geq 2\,\epsilon, \tag{7}$$

since otherwise $V_S = \emptyset$. Clearly, $v < u$ for all $v \in V(0, \xi, U)$ and all $u \in U(\xi + \epsilon, 1, U)$. (Here we need the assumption $\epsilon|U| \geq 2$.) Thus, by the definition of V_S, $e(U(\xi + \epsilon, 1, U), \{v\}) \leq (1 - \xi - 2\,\epsilon)\,d$ for all $v \in V_S$. Therefore,

$$\begin{aligned} e(U(\xi + \epsilon, 1, U), V_S) &\leq (1 - \xi - 2\,\epsilon)\,d\,|V_S| \\ &< (1 - \epsilon)\,(1 - \xi - \epsilon)\,d\,|V_S| \\ &\leq (1 - \epsilon)\,d\,\frac{|U(\xi + \epsilon, 1, U)|}{|U|}\,|V_S|, \end{aligned} \tag{8}$$

where the last inequality follows from

$$|U(\xi + \epsilon, 1, U)| = |U| - \lceil(\xi + \epsilon)|U|\rceil + 1 \geq (1 - \xi - \epsilon)|U|. \tag{9}$$

On the other hand, inequalities (7) and (9) imply $|U(\xi + \epsilon, 1, U)| \geq \epsilon|U|$. Hence, by Lemma 2.1 and inequality (8), we conclude $|V_S| \leq \epsilon|U|$, establishing the first claim in the lemma.

The second claim is entirely symmetric to the first and can be proved in exactly the same way.

We first observe that, for any $u \in U$ and any $V' \subseteq V$ such that $|V'| \geq \epsilon|U|$,

$$e(\{u\}, V') \leq d\,|V'|/|U| + \epsilon\,d\,|V'| < 2\,\epsilon\,d\,|V'|, \tag{10}$$

where the first inequality follows from Corollary 2.2 and the second inequality follows from $\epsilon|U| \geq 2$.

We now prove the third claim in the lemma by contradiction. Assume that there exist $X \subseteq V(0, \zeta, U)$ and $Y \subseteq V(\xi, 1, U)$ such that $\xi - \zeta \geq 6\epsilon$, $|X| \geq \epsilon|U|$, $|Y| \geq \epsilon|U|$, and SMALL$(x) \leq$ SMALL(y) for all $x \in X$ and all $y \in Y$. For all $x \in X \subseteq V(0, \zeta, U)$, SMALL$(x)$ is incremented by 1 whenever x is compared with any $u \in U(\zeta + \epsilon, 1, U)$. (Here we need the assumption $\epsilon|U| \geq 2$). In addition, for all $y \in Y \subseteq (\xi, 1, U)$, if SMALL$(y)$ is incremented at all after y is compared with an element $u \in U$, then $u \in U(\xi, 1, U)$. Therefore, for all $x \in X$ and all $y \in Y$, $e(U(\zeta + \epsilon, 1, U), \{x\}) \leq$ SMALL$(x) \leq$ SMALL$(y) \leq e(U(\xi, 1, U), \{y\})$. Hence,

$$e(U(\zeta + \epsilon, 1, U), X)/|X| \leq e(U(\xi, 1, U), Y)/|Y|. \tag{11}$$

By the fact $|U(0,\xi,U) \cap U(\xi,1,U)| \leq 1$ and inequality (10), we know that the right-hand side of inequality (11) is strictly less than $d - e(U(0,\xi,U),Y)/|Y| + 2\,\epsilon\,d$. Therefore,

$$e(U(\zeta+\epsilon,1,U),X)/|X| + e(U(0,\xi,U),Y)/|Y| < d + 2\epsilon d. \tag{12}$$

On the other hand, since $\zeta + \epsilon \leq \xi - 5\epsilon \leq 1$,

$$|U(\zeta+\epsilon,1,U)| = |U| - \lceil(\zeta+\epsilon)|U|\rceil + 1 \geq (1-\zeta-\epsilon)|U|. \tag{13}$$

Similarly,

$$|U(0,\xi,U)| = \lfloor \xi|U|\rfloor \geq \xi|U| - 1 \geq (\xi-\epsilon)|U|. \tag{14}$$

By the assumption $\xi - \zeta \geq 6\epsilon$, we immediately see that the right-hand sides of inequalities (13) and (14) are both at least $5\epsilon|U|$. Thus, applying Lemma 2.1 to inequality (12), we obtain $(1-\zeta-\epsilon-\epsilon)\,d + (\xi-\epsilon-\epsilon)\,d < d + 2\epsilon d$, contradicting the assumption $\xi - \zeta \geq 6\epsilon$. □

2.2.2. Construction of $S(X)$. $S(X)$ consists of three subsets $S_L(X)$, $S_R(X)$, and $S_C(X)$. In order to partition X properly, not only do we need to know $S(X)$ but we also need to know $S_L(X)$, $S_R(X)$, and $S_C(X)$. This subsection is devoted to constructing these sets.

We first introduce some concepts. Some of these concepts are not directly used in the construction of $S(X)$, but they are useful in understanding the relevant terminologies and analyzing our final algorithm. So we define these concepts here for ease of reference. The concepts of a natural interval and strangeness were used in [9].

- The *natural interval* of an AKS tree node is inductively defined as follows: the natural interval of the root of an AKS tree is $[1,n]$; if the natural interval of an AKS tree node X is $[\alpha,\beta]$, then the natural intervals of the left and right children of X are $[\alpha, \frac{\alpha+\beta-1}{2}]$ and $[\frac{\alpha+\beta+1}{2}, \beta]$, respectively.
- Let $[\alpha(X),\beta(X)]$ denote the natural interval of an AKS tree node X, and let $m(X) = \frac{\alpha(X)+\beta(X)}{2}$.
- The *strangeness* of an element x with respect to an AKS tree node X is defined to be the number of levels that x needs to move from X upward in X's AKS tree in order to reach the first AKS tree node whose natural interval contains $\mathrm{rank}(x)$. (Note that the strangeness of x with respect to X is well defined even if x is not located in X.)
- For each AKS tree node X, let $h(X)$ denote the *height* of X in its AKS tree. (The height of a leaf is assumed to be zero.)

CLAIM 2.1. *If X is an AKS tree node such that $h(X) \geq 0$ (i.e., X is either above or included in the leaf level), then* $\mathrm{cap}(X) \leq 6^{2-h(X)}\,(\beta(X) - \alpha(X) + 1)$.

Proof. Assume that X is at level i for some $i \geq 0$; i.e., X is i levels below the root. Consider the lowest level where each AKS tree node is full. This level is at least $h(X) - 2$ levels below X's level, since either a leaf is full or its grandparent is full. The sum of the capacities of all the nodes at the lowest full level is at most n, since there are at most n elements in an AKS tree. Hence,

$$2^{i+h(X)-2}\mathrm{cap}(X)\,A^{h(X)-2} \leq n = 2^i\,(\beta(X)-\alpha(X)+1)\,,$$

where the last equality holds since the sum of the natural-interval sizes at any level of an AKS tree is equal to n. The correctness of the claim follows immediately from the above inequality. □

In the next claim and the rest of the paper, we will use parameter c to denote a certain large integer constant. We will not give an explicit value of c, but we will see that a sufficiently large c will be good for our algorithm.

CLAIM 2.2. *If* $h(Y) \geq 0.5\,h(X) + c$ *and* $h(X) \geq 0$, *then* $\mathrm{cap}(X) < \beta(Y) - \alpha(Y) + 1$.

Proof. Note that $h(Y) \geq \frac{1}{2}h(X) + c$ implies $h(X) - h(Y) \leq \frac{1}{2}h(X) - c$. Hence, by Claim 2.1,

$$\begin{aligned}\mathrm{cap}(X) &\leq 6^{2-h(X)}\,(\beta(X) - \alpha(X) + 1)\\ &= 6^{2-h(X)}\,(\beta(Y) - \alpha(Y) + 1)\,2^{h(X)-h(Y)}\\ &\leq 6^{2-h(X)}\,(\beta(Y) - \alpha(Y) + 1)\,2^{\frac{1}{2}h(X)-c}\\ &< \beta(Y) - \alpha(Y) + 1,\end{aligned}$$

where the last inequality holds since c is sufficiently large and $h(X) \geq 0$. □

CLAIM 2.3. *For any* X *such that* $h(X) \geq 0$, *at each level with height at least* $0.5\,h(X) + c$ *in either* T_N *or* T_B, *there exists a unique AKS tree node whose natural interval contains* $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36 \rceil - 1]$.

Proof. Since natural intervals at the same level of an AKS tree cannot overlap with each other, we only need to show the existence of a desired node at each level. Moreover, since the natural interval of a node is contained in the natural interval of its parent, we only need to consider the level with height exactly $\lceil 0.5\,h(X) + c \rceil$. If $h(X) \leq 0.5\,h(X)+c$, then the ancestor of X with height $\lceil 0.5\,h(X) + c \rceil$ has the desired property since Claim 2.1 implies that X's natural interval contains $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36 \rceil - 1]$. If $h(X) > 0.5\,h(X) + c$, then let Y be the unique descendant of X at level $\lceil 0.5\,h(X) + c \rceil$ such that $\alpha(Y) = \alpha(X)$. By Claim 2.2, Y has the desired property. □

By Claim 2.3, the following notation of $X'_{L,i}$ $(i = 0, 1)$ is well defined. Similarly, we can verify that $X'_{R,i}\,(i = 0, 1)$, $X'_{C,0}$, $X'_{CL,1}$, and $X'_{CR,1}$ are all well defined. For an AKS tree node X in T_N (resp., T_B) such that $h(X) \geq 0$, let

- X' be the unique AKS tree node in T_B (resp., T_N) such that $[\alpha(X'), \beta(X')] = [\alpha(X), \beta(X)]$,
- $X'_{L,i}$ $(i = 0, 1)$ be the unique AKS tree node in T_B (resp., T_N) such that $h(X'_{L,i}) = \lceil 2^{-i}h(X')+c \rceil$ and $[\alpha(X'_{L,i}), \beta(X'_{L,i})] \supseteq [\alpha(X'), \alpha(X')+\lceil \mathrm{cap}(X')/36 \rceil - 1]$,
- $X'_{R,i}$ $(i = 0, 1)$ be the unique AKS tree node in T_B (resp., T_N) such that $h(X'_{R,i}) = \lceil 2^{-i}h(X') + c \rceil$ and $[\alpha(X'_{R,i}), \beta(X'_{R,i})] \supseteq [\beta(X') - \lceil \mathrm{cap}(X')/36 \rceil + 1, \beta(X')]$,
- $X'_{C,0}$ be the unique AKS tree node in T_B (resp., T_N) such that $h(X'_{C,0}) = h(X') + c$ and $[\alpha(X'_{C,0}), \beta(X'_{C,0})] \supseteq [m(X') - \lceil \mathrm{cap}(X')/72 \rceil + 1/2, m(X') + \lceil \mathrm{cap}(X')/72 \rceil - 1/2]$,
- $X'_{CL,1}$ be the unique AKS tree node in T_B (resp., T_N) such that $h(X'_{CL,1}) = \lceil 2^{-1}h(X') + c \rceil$ and $[\alpha(X'_{CL,1}), \beta(X'_{CL,1})] \supseteq [m(X') - \lceil \mathrm{cap}(X')/72 \rceil + 1/2, m(X') - 1/2]$,
- $X'_{CR,1}$ be the unique AKS tree node in T_B (resp., T_N) such that $h(X'_{CR,1}) = \lceil 2^{-1}h(X')+c \rceil$ and $[\alpha(X'_{CR,1}), \beta(X'_{CR,1})] \supseteq [m(X')+1/2, m(X')+\lceil \mathrm{cap}(X')/72 \rceil$

$-1/2]$,

- $P_L(X)$ be the unique path from $X'_{L,0}$ to $X'_{L,1}$,
- $P_R(X)$ be the unique path from $X'_{R,0}$ to $X'_{R,1}$,
- $P_{CL}(X)$ be the unique path from $X'_{C,0}$ to $X'_{CL,1}$,
- $P_{CR}(X)$ be the unique path from $X'_{C,0}$ to $X'_{CR,1}$.

In the above definition, $P_L(X)$ is assumed to contain the nodes $X'_{L,0}$ and $X'_{L,1}$. Similarly, each of the other three paths ($P_R(X)$, $P_{CL}(X)$, $P_{CR}(X)$) contains its end nodes described above. We remark that the left and right ends of each interval in the above definitions are integers. In particular, to ensure that $m(X)-1/2$ is an integer, we may assume without loss of generality $\alpha(X) \neq \beta(X)$, since we will only need to partition an AKS tree node X with $\alpha(X) \neq \beta(X)$. (If $\alpha(X) = \beta(X)$, then X can only contain one element and there is no need to partition X.)

We are now ready to define $S_L(X)$, $S_R(X)$, and $S_C(X)$.

- Let T_X denote the subtree of X's AKS tree that is rooted at X, and let $T_X(d)$ denote the subtree of T_X consisting of all nodes in T_X that are d levels within X. (Note that $T_X(d)$ contains exactly $d+1$ levels.)
- Let

$$S_L(X) = \bigcup_{Y' \in P_L(X)} T_{Y'}\left(\lceil 0.5\, h(X') + c \rceil\right),$$

$$S_R(X) = \bigcup_{Y' \in P_R(X)} T_{Y'}(\lceil 0.5\, h(X') + c \rceil), \text{ and}$$

$$S_C(X) = \bigcup_{Y' \in P_{CL}(X) \cup P_{CR}(X)} T_{Y'}(\lceil 0.5\, h(X') + c \rceil).$$

Note that $S_L(X)$, $S_R(X)$, and $S_C(X)$ are supposed to be sets of bolts or nuts, but the above definitions define them as sets of AKS tree nodes. As we have used X to denote both an AKS tree node and the list of elements in X, we use $S_L(X)$, $S_R(X)$, and $S_C(X)$ to denote both the sets of AKS tree nodes as defined above and the lists of nuts or bolts contained therein, as long as the meaning is clear from the context. Roughly, $S_L(X)$ (resp., $S_R(X)$) looks like a "tape" attached to path $P_L(X)$ (resp., $P_R(X)$). The tape has a "width" of $\lceil 0.5\, h(X) + c \rceil$ and vertically extends from c levels above X' all the way down to the leaf level. Similarly, $S_C(X)$ looks like two tapes of a similar shape. As a direct consequence, we know that

$$|S(X)| \geq |X|, \tag{15}$$

which will be used in the proof of Claim 3.7. The intuition behind this complicated definition of $S_L(X)$, $S_R(X)$, and $S_C(X)$ will become clear in the proof of Theorem 3.1.

2.2.3. A partition algorithm. In this subsection, we describe how to partition X into FL, CL, CR, and FR by comparing elements in X with elements in $S(X)$, according to Algorithm COMPARE described in subsection 2.2.1. The reason that our algorithm can provide a proper partition of X is fairly lengthy and will be discussed in section 3. In particular, it is dependent upon another key property of the original AKS sorting algorithm.

Note that Lemma 2.3 only states that Algorithm COMPARE(U, V, G) sometimes gives a proper partition of V, the larger set between U and V. In fact, a careful investigation of the proof of Lemma 2.3 reveals that when V is substantially larger

than U, not much can be said about the ranking of U (the smaller set between U and V) by COMPARE(U, V, G). On the other hand, however, we will need to partition X by comparing X with $S(X)$, which can be much larger than X in many cases. Hence, in the most interesting case (see Case 2 below), the following algorithm, PARTITION(X), consists of two major phases. In the first phase, we choose subsets $S'_L(X) \subseteq S_L(X)$, $S'_C(X) \subseteq S_C(X)$, and $S'_R(X) \subseteq S_R(X)$, each of which has size comparable to $|X|$, and we let $S'(X) = S'_L(X) \cup S'_C(X) \cup S'_R(X)$. In the second phase, we use $S'(X)$ to partition X into FL, CL, CR, and FR.

ALGORITHM PARTITION(X).

Let ϵ be a sufficiently small constant. There are two cases.

Case 1. $\epsilon|X| < 2$. We compare all elements in X with all elements in $S(X)$. Then, we construct a graph on all elements of X by drawing a directed edge from $x_1 \in X$ to $x_2 \in X$ if there exists an element $x \in S(X)$ such that $x_1 \le x \le x_2$. Such a graph is a directed acyclic graph (DAG), and we can topologically sort X according to the DAG. According to this order, we divide X into FL, CL, CR, and FR, each with size specified in the AKS sorting algorithm, i.e., $|FL| = |FR| = \min\{\frac{\lambda}{2}\operatorname{cap}(X), \frac{|X|}{2}\}$, and $|CL| = |CR| = \frac{|X|}{2} - |FL|$.

Case 2. $\epsilon|X| \ge 2$. Let G be a bipartite graph described in Lemma 2.1. In particular, in the first three steps of the algorithm, $G \subseteq X \times S_L(X)$, $G \subseteq X \times S_R(X)$, and $G \subseteq X \times S_C(X)$, resp., and in the last step of the algorithm, $G \subseteq S'(X) \times X$. As we will see in the proof of Theorem 3.1, θ will be a fraction of λ for $G \subseteq S'(X) \times X$.

Step 1. Apply COMPARE$(X, S_L(X), G)$. Let $S'_L(X)$ consist of $\lceil(\lambda/10)|X|\rceil$ elements in $S_L(X)$ with the smallest SMALL values among those whose SMALL values are at least $d\,(|X| - \frac{2\lambda}{5}\operatorname{cap}(X) - 2\,\epsilon\,|X|)/|X|$. (Ties are broken arbitrarily.)

Step 2. Apply COMPARE$(X, S_R(X), G)$. Let $S'_R(X)$ consist of $\lceil(\lambda/10)|X|\rceil$ elements in $S_R(X)$ with the smallest LARGE values among those whose LARGE values are at least $d\,(|X| - \frac{2\lambda}{5}\operatorname{cap}(X) - 2\,\epsilon\,|X|)/|X|$. (Ties are broken arbitrarily.)

Step 3. Apply COMPARE$(X, S_C(X), G)$. Let $S'_{CL}(X)$ consist of at most $\lceil(1/2 - \lambda/10)|X|\rceil$ elements in $S_C(X)$ with the smallest SMALL values among those whose SMALL values are at least $(1/2 - 2\epsilon)d$. (That is, (i) if there are more than $\lceil(1/2 - \lambda/10)|X|\rceil$ elements in $S_C(X)$ having their SMALL values at least $(1/2 - 2\epsilon)d$, then let $S'_{CL}(X)$ consist of $\lceil(1/2 - \lambda/10)|X|\rceil$ elements in $S_C(X)$ with the smallest SMALL values among those whose SMALL values are at least $(1/2 - 2\epsilon)d$; and (ii) if there are fewer than $\lceil(1/2 - \lambda/10)|X|\rceil$ elements in $S_C(X)$ having their SMALL values at least $(1/2 - 2\epsilon)d$, then let $S'_{CL}(X)$ consist of all these elements.) Similarly, let $S'_{CR}(X)$ consist of at most $\lceil(1/2 - \lambda/10)|X|\rceil$ elements in $S_C(X)$ with the smallest LARGE values among those whose LARGE values are at least $(1/2 - 2\epsilon)d$. (Ties are broken arbitrarily.)

Include all elements in $S'_{CL}(X)$ and $S'_{CR}(X)$ into $S'_C(X)$. If $S'_C(X)$ now has fewer than $\lceil(1 - \lambda/5)|X|\rceil$ elements, put an additional arbitrary set of elements from $S_C(X)$ into $S'_C(X)$ so that $S'_C(X)$ contains exactly $\lceil(1 - \lambda/5)|X|\rceil$ elements.

Step 4. Let $S'(X) = S'_L(X) \bigcup S'_R(X) \bigcup S'_C(X)$. Apply COMPARE$(S'(X), X, G)$. Use COUNTINGSORT to sort all elements in X according to their SMALL values, with the element with the largest SMALL value listed first. (Ties are broken arbitrarily.) According to this order, we divide X (from the first to the last) into FL, CL, CR, and FR, each with size specified in the AKS sorting algorithm.

Remark. Since $|S_C(X)| \ge |P_{CL}(X)| \ge |X|$, where $|S_C(X)|$ (resp., $|P_{CL}(X)|$)

denotes the number of elements contained in $S_C(X)$ (resp., $P_{CL}(X)$), there are always enough (i.e., $\lceil (1 - \lambda/5)|X| \rceil$) elements to be included in $S'_C(X)$ in Step 3. However, it is not clear at all why there are always enough (i.e., $\lceil (\lambda/10)|X| \rceil$) elements to be included in $S'_L(X)$ or $S'_R(X)$ in Steps 1 or 2. But we will show in the proof of Theorem 3.1 that there are always sufficiently many elements to be included in $S'_L(X)$ and $S'_R(X)$ when we use PARTITION(X) within our final algorithm for sorting nuts and bolts (see inequality (23)).

3. An analysis of the algorithm. In this section, we sketch the correctness proof and the running-time analysis of our algorithm for sorting nuts and bolts.

THEOREM 3.1. *The algorithm described in the preceding section sorts n nuts and n bolts in $O(n \log n)$ time.*

Proof. Our proof consists of two parts: the first part deals with the correctness of the algorithm, and the second part deals with the running time of the algorithm.

Proof of correctness. In this part of the proof, we first prove a few claims concerning certain properties of the sets $S_L(X)$, $S_R(X)$, $S_C(X)$, $S'_L(X)$, $S'_R(X)$, and $S'_C(X)$. These claims provide intuition as to why $S'(X) = S'_L(X) \cup S'_R(X) \cup S'_C(X)$ contains necessary elements for partitioning X by using the graph of Lemma 2.1.

We first introduce some notation. Let

$$\eta = 2\mu\delta \frac{A^2}{1 - 4\delta^2 A^2} + \frac{1}{8A^2 - 2A} + \mu.$$

Note that our η is (slightly) different from the parameter η defined in [9, p. 86], but they play a similar role in the analyses. In particular, according to the analysis of [9], η bounds the unbalance of an AKS tree node X in the following sense. Name the children of X as X_1 and X_2. It is possible that more than half of the elements contained in X may have ranks belonging to the natural interval of one of its children, say, X_1. In such a case, when X is partitioned into left and right halves (i.e., $FL \cup CL$ and $FR \cup CR$), some elements with ranks belonging to X_1 may be moved into X_2 (even if the partition of X is done perfectly) simply because of the capacity constraint on X_1. This will cause some of the elements moved into X_2 to become strange with respect to the node that they reside in for the first time. The analysis of [9] shows that the number of elements that will be forced into X_2 by such a reason is at most $\eta \operatorname{cap}(X)$ (see Claim 3.1). According to the current parameter choices, we have

$$\eta = \frac{8627}{154836} = 0.0557\ldots.$$

As discussed in subsection 2.1, we will not specify the value of the parameter ε, which is used in [9]. Rather, we assume ε to be a sufficiently small constant. Correspondingly, we assume that graph G used in Algorithm PARTITION has parameter ϵ as a sufficiently small constant depending upon ε and has parameter θ as a properly chosen constant, which is a fraction of λ. Finally, we define $S_r(X)$ to be the number of elements that are contained in X and are r or more strange with respect to X. Note that our definition of $S_r(X)$ is slightly different from that of [9], in which $S_r(X)$ is defined as the ratio of the quantity in our definition to $\operatorname{cap}(X)$.

We will establish the correctness of the algorithm by proving that the following two properties hold throughout the execution of the algorithm.

PROPERTY 3.1. *For any AKS tree node X and for any $r \geq 1$,*

$$S_r(X) \leq \mu\, \delta^{r-1} \operatorname{cap}(X). \tag{16}$$

PROPERTY 3.2. *For any $r \geq 1$ and any AKS tree node X such that $|X| \geq \lambda \operatorname{cap}(X)$, when Algorithm* PARTITION(X) *is executed,* (1) *at most $\varepsilon\mu\delta^{r-1}\operatorname{cap}(X)$ elements in X whose strangeness with respect to X is r or more can be placed into $CL \cup CR$;* (2) *at most $(\eta+\varepsilon)\operatorname{cap}(X)$ elements in X whose ranks are at most $m(X)$ can be placed into CR; and* (3) *at most $(\eta+\varepsilon)\operatorname{cap}(X)$ elements whose ranks are at least $m(X)$ can be placed into CL.*

Before proving these properties, we point out that, as in [9], Property 3.1 alone is sufficient to establish the correctness of the algorithm, since, toward the end of the algorithm when $\operatorname{cap}(X)$ is less than a sufficiently small constant for all nonempty X, Property 3.1 implies that no item can be strange with respect to the AKS tree node that it resides in. In [9], Property 3.1 is difficult to prove, and an analogue of Property 3.2 was relatively easily verified by the so-called ε-halver property, which in turn depends on expander graphs. More precisely, an analogue of Property 3.2 in [9] is independent of Property 3.1, except that the analogues to the second and the third statements in Property 3.2 are dependent on Claim 3.1, which, in turn, depends on Property 3.1. In our algorithm, however, the two properties are highly dependent on each other. In particular, Algorithm PARTITION would not provide a reasonable partition of X without the validity of Property 3.1 because we cannot always keep a match between X and X'. Thus, we will need to prove both properties simultaneously in the analysis of our algorithm, whereas the analysis given in [9] for the original AKS sorting algorithm only needs to focus on the proof of Property 3.1.

We now start proving Properties 3.1 and 3.2. The analysis of the original AKS sorting algorithm provided in [9] actually shows that inequality (16) always holds as long as Property 3.2 is never violated. This means that if either Property 3.1 or Property 3.2 is violated at all, Property 3.2 must be violated first. Hence, it suffices to show that Property 3.2 always holds as long as Property 3.1 always holds. In particular, we prove Property 3.2 under the following assumption.

Assumption 3.1. Inequality (16) always holds.

Given Assumption 3.1, the following claim is proved in [9].

CLAIM 3.1. *An AKS tree node X contains at most $|X|/2 + \eta \operatorname{cap}(X)$ elements with ranks more than $m(X)$ and at most $|X|/2 + \eta \operatorname{cap}(X)$ elements with ranks less than $m(X)$.*

Proof. See [9, pp. 80–81]. □

In what follows, we prove a few claims for an arbitrary AKS tree node X.

CLAIM 3.2. *For $r \geq 1$, T_X contains at most $\frac{\mu\,\delta^{r-1}}{1-2\,A\,\delta} \operatorname{cap}(X)$ elements that are r or more strange with respect to X.*

Proof. If an element is r or more strange with respect to X, it must be $r+d$ or more strange with respect to a node that is d levels below X in T_X. Hence, by Assumption 3.1, the number of elements that are r or more strange with respect to X and are located exactly d levels below X in T_X is upper bounded by $\mu\,\delta^{r+d-1}\,2^d A^d \operatorname{cap}(X)$. By summing these quantities over all $d \geq 0$, we obtain the correctness of the claim. □

CLAIM 3.3. *For any $r \geq c+1$, where c is the constant described immediately before Claim* 2.2, *$S_L(X)$, $S_R(X)$, and $S_C(X)$ contain at most $\frac{\mu\,\delta^{r-c-1}}{(1-2\,A\,\delta)\,A^c} \operatorname{cap}(X)$, $\frac{\mu\,\delta^{r-c-1}}{(1-2\,A\,\delta)\,A^c} \operatorname{cap}(X)$, and $\frac{2\mu\,\delta^{r-c-1}}{(1-2\,A\,\delta)\,A^c} \operatorname{cap}(X)$, resp., elements whose strangeness with respect to X' is at least r.*

Proof. We only prove the claim for $S_L(X)$. The cases of $S_R(X)$ and $S_C(X)$ can be proved in the same fashion. By definitions, all of the elements in $S_L(X)$ are located in $T_{X'_{L,0}}$ (see subsection 2.2.2 for the definition of $X'_{L,0}$). Moreover, $\mathrm{cap}(X'_{L,0}) = \frac{\mathrm{cap}(X)}{A^c}$, since $X'_{L,0}$ is c levels above X'. Now, the correctness of the claim for $S_L(X)$ follows from Claim 3.2. □

In the next claim, ε_1 is an arbitrarily small constant, which will be much smaller than ε. This is obtained at the cost of making c be a sufficiently large constant and making ϵ of Lemma 2.1 be a sufficiently small constant, much smaller than ε_1. For example, see inequality (34) and the argument immediately after inequality (28). In both places, we need to assume ϵ/ε and/or $\varepsilon_1/\varepsilon$ to be sufficiently small.

CLAIM 3.4. (1) $S_L(X)$ *contains all but at most* $\varepsilon_1 \,\mathrm{cap}(X)/36$ *of the elements whose ranks are in* $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36 \rceil - 1]$*;* (2) $S_R(X)$ *contains all but at most* $\varepsilon_1 \,\mathrm{cap}(X)/36$ *of the elements whose ranks are in* $[\beta(X) - \lceil \mathrm{cap}(X)/36 \rceil + 1, \beta(X)]$*;* (3) $S_C(X)$ *contains all but at most* $\varepsilon_1 \,\mathrm{cap}(X)/36$ *of the elements whose ranks are in* $[m(X) - \lceil \mathrm{cap}(X)/72 \rceil + 1/2, m(X) + \lceil \mathrm{cap}(X)/72 \rceil - 1/2]$.

Proof. We only prove the claim for $S_L(X)$. The cases of $S_R(X)$ and $S_C(X)$ can be proved in the same fashion. Without loss of generality, we assume that $X \in T_B$. For ease of notation, we call a bolt *good* if its rank (in N or B) is in $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36 \rceil - 1]$. We need to prove that $T_B - S_L(X)$ contains at most $\varepsilon_1 \,\mathrm{cap}(X)/36$ good bolts. Since $S_L(X) \subseteq T_{X'_{L,0}} \subseteq T_B$, $T_B - S_L(X)$ consists of two parts: $T_B - T_{X'_{L,0}}$ and $T_{X'_{L,0}} - S_L(X)$. We will show that each of the two parts contains at most $\varepsilon_1 \,\mathrm{cap}(X)/72$ good bolts.

Let

$$P = \text{the path from } X'_{L,0} \text{ to the root of } T_B, \text{ including both } X'_{L,0} \text{ and the root,}$$
$$\bar{P} = \{Y' \in T_B - T_{X'_{L,0}} - P \mid \text{ the parent of } Y' \text{ is in } P\}.$$

Clearly, $\bar{P}$ contains exactly one AKS tree node at each level above (*inclusively*) the level of $X'_{L,0}$ and below (*exclusively*) the root level. Thus, we can list all of the nodes in $\bar{P}$ from bottom to top as $Y'_1, Y'_2, \ldots, Y'_{|\bar{P}|}$. Clearly, $T_B - T_{X'_{L,0}} - P$ can be partitioned as

$$T_B - T_{X'_{L,0}} - P = \bigcup_{1 \le i \le |\bar{P}|} T_{Y'_i}. \tag{17}$$

Moreover, a good bolt y in $T_{Y'_i}$ has $\mathrm{rank}(y)$ belonging to the natural interval of the sibling of Y'_i (which is in P), so, by Claim 2.3, y must be at least 1-strange with respect to Y'_i. Thus, by Claim 3.2, the number of good bolts contained in $T_{Y'_i}$ is at most

$$\frac{\mu}{1 - 2\,A\,\delta} \,\mathrm{cap}(Y'_i) \le \frac{\mu}{1 - 2\,A\,\delta} \,\frac{\mathrm{cap}(X)}{A^{c+i-1}}.$$

Summing up the above quantities over all $i \ge 1$, we know by (17) that the total number of good bolts contained in $T_B - T_{X'_{L,0}} - P$ is at most

$$\frac{\mu}{1 - 2\,A\,\delta} \,\frac{\mathrm{cap}(X)}{A^c(1 - \frac{1}{A})}.$$

On the other hand, the total number of good bolts contained in P is at most the total capacity of the nodes along path P, which can be upper bounded by

$$\frac{\mathrm{cap}(X)}{A^c(1-\frac{1}{A})}.$$

Summing the preceding two terms, we know that the number of good bolts contained in $T_B - T_{X'_{L,0}}$ is at most

$$\left(1+\frac{\mu}{1-2\,A\,\delta}\right)\frac{\mathrm{cap}(X)}{A^c(1-\frac{1}{A})} \le \frac{\varepsilon_1\,\mathrm{cap}(X)}{72}, \tag{18}$$

where the inequality holds since c is assumed to be a sufficiently large constant.

We next show that $T_{X'_{L,0}} - S_L(X)$ contains at most $\varepsilon_1\,\mathrm{cap}(X)/72$ good bolts. List all of the nodes in $P_L(X)$ (see subsection 2.2.2 for the definition) from top to bottom as

$$Z'_1, Z'_2, \ldots, Z'_{\lfloor 0.5\,h(X)\rfloor+1},$$

where $Z'_1 = X'_{L,0}$ and $Z'_{\lfloor 0.5\,h(X)\rfloor+1} = X'_{L,1}$. For each i such that $1 \le i \le \lfloor 0.5\,h(X)\rfloor + 1$, there are exactly

$$J = 2^{\lceil 0.5\,h(X)+c\rceil+1}$$

nodes in $T_{X'_{L,0}}$ that are descendants of Z_i and have distance exactly $\lceil 0.5\,h(X)+c\rceil+1$ from Z'_i. Some of these elements are located within $T_{X'_{L,0}} - S_L(X)$, and we list all of them as

$$Z'_{i,1}, Z'_{i,2}, \ldots, Z'_{i,J(i)},$$

where $J(i) \le J$. For example, $J(\lfloor 0.5h(X)\rfloor + 1) = 2^{\lceil 0.5h(X)+c\rceil+1}$. Clearly,

$$T_{X'_{L,0}} - S_L(X) \subseteq \bigcup_{1\le i\le \lfloor 0.5\,h(X)\rfloor+1}\ \bigcup_{1\le j\le J(i)} T_{Z'_{i,j}}. \tag{19}$$

By the definition of $P_L(X)$ and $S_L(X)$, if a node is in $T_{X'_{L,0}}$ and has its natural interval overlapping with $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36\rceil - 1]$, then the node must be in $S_L(X)$. Since $Z'_{i,j} \notin S_L(X)$ for all $1 \le j \le J(i)$, each $Z'_{i,j}$ has its natural interval non-overlapping with $[\alpha(X), \alpha(X) + \lceil \mathrm{cap}(X)/36\rceil - 1]$. On the other hand, by definition, the rank of a good bolt is contained in $[\alpha(X),\ \alpha(X) + \lceil \mathrm{cap}(X)/36\rceil - 1]$. Thus, the strangeness of a good bolt with respect to $Z'_{i,j}$ must be at least $\lceil 0.5\,h(X) + c\rceil + 1$, since the bolt needs to move up to $Z_i \in P_L(X)$ to become nonstrange for the first time. Thus, by Claim 3.2, the number of good bolts in $T_{Z'_{i,j}}$ is at most

$$\begin{aligned}\frac{\mu\,\delta^{0.5\,h(X)+c}}{1-2\,A\,\delta}\,\mathrm{cap}(Z'_{i,j}) &\le \frac{\mu\,\delta^{0.5\,h(X)+c}}{1-2\,A\,\delta}\,\mathrm{cap}(Z'_i)\,A^{0.5\,h(X)+c+1}\\ &\le \frac{\mu\,\delta^{0.5\,h(X)+c}}{1-2\,A\,\delta}\,\mathrm{cap}(X)\,A^{h(X)+c+1}.\end{aligned}$$

Summing up this quantity over $1 \le i \le \lfloor 0.5\,h(X)\rfloor + 1$ and $1 \le j \le J(i) \le J$, we know by inequality (19) that the number of good bolts contained in $T_{X'_{L,0}} - S_L(X)$ is at most

$$(0.5\,h(X)+1)\,2^{\lceil 0.5\,h(x)+c\rceil+1}\,\frac{\mu\,\delta^{0.5\,h(X)+c}}{1-2\,A\,\delta}\,\mathrm{cap}(X)\,A^{h(X)+c+1} \le \frac{\varepsilon_1}{72}\mathrm{cap}(X),$$

where the inequality holds since c is assumed to be a sufficiently large constant. This, together with inequality (18), proves the first statement of the claim. □

We are now ready to prove Property 3.2. We first consider the simple case where $\epsilon|X| < 2$. Assume for contradiction that $S_L(X)$ contains no element with rank in $[\alpha(X),\ \alpha(X) + \varepsilon_1 \operatorname{cap}(X)]$. Then none of the $\min\{\lceil \operatorname{cap}(X)/36 \rceil,\ \lfloor \varepsilon_1 \operatorname{cap}(X) \rfloor + 1\}$ elements with ranks in $[\alpha(X),\ \alpha(X) + \min\{\lceil \operatorname{cap}(X)/36 \rceil - 1,\ \lfloor \varepsilon_1 \operatorname{cap}(X) \rfloor\}]$ is in $S_L(X)$. Thus, by Claim 3.4, $\min\{\lceil \operatorname{cap}(X)/36 \rceil,\ \lfloor \varepsilon_1 \operatorname{cap}(X) \rfloor + 1\} \leq \varepsilon_1 \operatorname{cap}(X)/36$. Therefore, $\min\{\operatorname{cap}(X)/36,\ \varepsilon_1 \operatorname{cap}(X)\} \leq \varepsilon_1 \operatorname{cap}(X)/36$, which is a contradiction. We have thus shown that $S_L(X)$ contains an element whose rank is in $[\alpha(X),\ \alpha(X) + \varepsilon_1 \operatorname{cap}(X)]$. Therefore, in the topologically sorted order found at the end of PARTITION(X), every element with rank less than or equal to $\alpha(X)$ is listed before every element with rank greater than or equal to $\alpha(X) + \lfloor \varepsilon_1 \operatorname{cap}(X) \rfloor$. The only elements that have ranks greater than or equal to $\alpha(X)$ but may be listed before elements with ranks strictly less than $\alpha(X)$ are those whose ranks are in $[\alpha(X),\ \alpha(X) + \lfloor \varepsilon_1 \operatorname{cap}(X) \rfloor - 1]$. For sufficiently small ε_1, we have $\mu \operatorname{cap}(X) + \varepsilon_1 \operatorname{cap}(X) < (\lambda/2) \operatorname{cap}(X)$. Thus, by Assumption 3.1, all of the $\mu \operatorname{cap}(X)$ or fewer elements of X that have ranks strictly smaller than $\alpha(X)$ will be caught into FL, establishing the first statement of Property 3.2. Using Assumption 3.1 and Claim 3.4, we can prove the second and third statements of Property 3.2 in a similar fashion (here we also need Claim 3.1).

In what follows, we prove Property 3.2 for the case $\epsilon|X| \geq 2$. The proof for this case is quite involved, and we prove the three statements of Property 3.2 one by one.

We begin with a proof for the first statement in Property 3.2. For any r, let

$$\bar{X}(r) = \{x \in X - FL \mid \operatorname{rank}(x) < \alpha(X) \text{ and } x \text{ is at least } r\text{-strange with respect to } X\ \}.$$

To prove the first statement in Property 3.2, it suffices to show

$$|\bar{X}(r)| \leq (\varepsilon/2)\mu\, \delta^{r-1} \operatorname{cap}(X). \tag{20}$$

(Similarly, we can show inequality (20) even if $\bar{X}(r)$ is defined to be the subset of $X - FR$ consisting of all elements that have ranks greater than $\beta(X)$ but are at least r-strange with respect to X.)

In what follows, let

$$IS_L(X) = \{x \in S_L(X) \mid \operatorname{rank}(x) \in [\alpha(X), \alpha(X) + (\lambda/9) \operatorname{cap}(X) - 1].$$

By the first statement of Claim 3.4 and $\lambda/9 \leq 1/36$,

$$|IS_L(X)| \geq \left\lfloor \frac{\lambda}{9} \operatorname{cap}(X) \right\rfloor - \varepsilon_1 \operatorname{cap}(X). \tag{21}$$

Clearly, for all $x \in IS_L(X)$,

$$S_1(X) \leq \operatorname{rank}(x, X) \leq S_1(X) + (\lambda/9) \operatorname{cap}(X). \tag{22}$$

Hence, by the first statement of Lemma 2.3, at most $\epsilon|X|$ elements of $IS_L(X)$ may have their SMALL values (after COMPARE$(X, S_L(X), G)$) less than or equal to $d\,(1 - (S_1(X) + (\lambda/9) \operatorname{cap}(X))/|X| - 2\epsilon)$. By inequality (21), this means that at least $\lfloor (\lambda/9) \operatorname{cap}(X) \rfloor - \varepsilon_1 \operatorname{cap}(X) - \epsilon|X| \geq (\lambda/10) \operatorname{cap}(X)$ elements of $IS_L(X)$ have their SMALL values greater than or equal to $d\,(1 - (S_1(X) + (\lambda/9) \operatorname{cap}(X))/|X| - 2\epsilon) \geq$

$d\,(|X|-(2\lambda/5)\,\mathrm{cap}(X)-2\epsilon|X|)/|X|$, where the inequality follows from Assumption 3.1. Thus, we have verified the remark for $S'_L(X)$ immediately after the description of PARTITION(X) by showing that there exist enough elements to be included in $S'_L(X)$, i.e.,

$$|S'_L(X)| \geq \frac{\lambda}{10}\mathrm{cap}(X). \tag{23}$$

Of course, $S'_L(X) \subseteq IS_L(X)$ may not hold, and inequality (22) may not hold for elements in $S'_L(X)$. However, we will show the following fact, which is an analogue to inequality (22).

CLAIM 3.5. *$S'_L(X)$ contains at least $(\lambda/10)\,\mathrm{cap}(X) - 2\epsilon|X|$ x's satisfying the following inequality:*

$$S_1(X) - 6\epsilon|X| \leq \mathrm{rank}(x, X) \leq (2\lambda/5)\mathrm{cap}(X) + 4\epsilon|X|. \tag{24}$$

Proof. According to the second statement of Lemma 2.3, at most $\epsilon|X|$ elements whose ranks with respect to X are greater than or equal to $(2\lambda/5)\,\mathrm{cap}(X)+4\epsilon|X|$ may have their LARGE values less than or equal to $d\,((2\lambda/5)\,\mathrm{cap}(X) + 2\epsilon|X|)/|X|$. This means that at most $\epsilon|X|$ elements violating the second inequality of inequality (24) may be included in $S'_L(X)$. On the other hand, since, in Step 1 of PARTITION(X), we have given priority to elements with the smallest SMALL values in choosing among all elements whose SMALL values are greater than or equal to $d\,(|X| - (2\lambda/5)\,\mathrm{cap}(X) - 2\epsilon|X|)/|X|$, by the third statement of Lemma 2.3, the second inequality in inequality (22), and the fact that $IS_L(X)$ has at least $(\lambda/10)\,\mathrm{cap}(X)$ elements with SMALL values greater than or equal to $d(|X| - (2\lambda/5)\,\mathrm{cap}(X) - 2\epsilon|X|)/|X|$ (see between inequalities (22) and (23)), at most $\epsilon|X|$ elements violating the first of inequality (24) may be included in $S'_L(X)$. Now, the claim follows from inequality (23). □

Let

$$\bar{Y}(r) = \left\{ y \in X \;\middle|\; \begin{array}{l} \mathrm{rank}(y, X) \geq (\lambda/2)\mathrm{cap}(X) \text{ and } y \text{ is listed before} \\ \text{all elements of } \bar{X}(r) \text{ at the end of PARTITION}(X) \end{array} \right\}.$$

Ideally, all elements in $\bar{X}(r)$ should be listed among the first $(\lambda/2)\,\mathrm{cap}(X)$ positions at the end of PARTITION(X). In reality, no element in $\bar{X}(r)$ is listed among the first $(\lambda/2)\,\mathrm{cap}(X)$ positions. Thus, among all the first $(\lambda/2)\,\mathrm{cap}(X)$ positions, up to $|\bar{X}(r)|$ positions that should ideally be occupied by elements in $\bar{X}(r)$ must be filled by elements in $\bar{Y}(r)$. Hence,

$$|\bar{Y}(r)| \geq |\bar{X}(r)|. \tag{25}$$

By $\epsilon|X| \geq 2$, we know that all elements x's satisfying inequality (24) are smaller than all elements in $\bar{Y}(r)$ and are larger than at least $|\bar{X}(r)| - 7\epsilon|X|$ elements in $\bar{X}(r)$ (by the definition of $S_1(X)$). Hence, by Claim 3.5, for all $y \in \bar{Y}(r)$, $\mathrm{rank}(y, S'_L(X)) - \mathrm{rank}(x, S'_L(X)) \geq (\lambda/10)\,\mathrm{cap}(X) - 2\,\epsilon|X| > 6\,\epsilon|X|$ for at least $|\bar{X}(r)| - 7\epsilon|X|$ elements x's in $\bar{X}(r)$. Thus, by the third statement of Lemma 2.3 and inequality (25), we conclude that $|\bar{X}(r)| - 7\epsilon|X| < \epsilon|X|$, which implies

$$\bar{X}(r) < 8\epsilon|X|. \tag{26}$$

This inequality holds for all $r \geq 1$, and it implies inequality (20) for all r such that $8\epsilon \leq \varepsilon\mu\delta^{r-1}$. So, in what follows, we will assume that r satisfies the inequality

$$\varepsilon\mu\delta^{r-1} < 8\epsilon. \tag{27}$$

By $\epsilon|X| \geq 2$, all the elements of X whose ranks with respect to X are at least $(2\lambda/5)\mathrm{cap}(X) + 5\epsilon|X|$ are strictly larger than all elements satisfying the second inequality in inequality (24). By Claim 3.5 and the second statement of Lemma 2.3, at the end of COMPARE$(S'_L(X), X)$ in PARTITION(X), at least $|X|-(2\lambda/5)\mathrm{cap}(X)-6\epsilon|X|$ elements of X whose ranks with respect to X are at least $(2\lambda/5)\mathrm{cap}(X)+5\epsilon|X|$ have LARGE values greater than or equal to $d(S'_L(X) - 2\epsilon|X| - 2\epsilon|X|)/|X| \geq (\lambda/11)d$, where the inequality follows from inequality (23). Thus, since the elements in $\bar{X}(r)$ are all listed after the first $(\lambda/2)\mathrm{cap}(X)$ of X at the end of PARTITION(X), the elements in $\bar{X}(r)$ all have LARGE values greater than or equal to $(\lambda/11)d$ at the end of COMPARE$(S'_L(X), X)$ of PARTITION(X).

Now applying the third statement of Lemma 2.1 with $\theta = \lambda/11$ to an arbitrary subset of $\bar{X}(r)$ with $\min\{|\bar{X}(r)|, 2\,e^{-4}\theta\,|X|/d\}$ elements (where d is the degree of each vertex in X in COMPARE$(S'_L(X), X, G)$), the elements of this subset are directly connected to at least $\min\{\theta d|\bar{X}(r)|/2,\, e^{-4}\theta^2|X|\}$ elements of $S'_L(X)$ by an edge whose corresponding comparison shows the element in $S'_L(X)$ is smaller than or equal to its opponent. This means that $S'_L(X)$ contains at least $\min\{\theta d|\bar{X}(r)|/2,\, e^{-4}\theta^2|X|\}$ elements that are smaller than or equal to at least one element in $\bar{X}(r)$ and are therefore at least r-strange with respect to X. Hence, by $S'_L(X) \subseteq S_L(X)$ and Claim 3.3,

$$\min\left\{\frac{\theta d|\bar{X}(r)|}{2},\, e^{-4}\theta^2|X|\right\} \leq \frac{\mu\,\delta^{r-c-1}}{(1-2\,A\,\delta)\,A^c}\,\mathrm{cap}(X). \tag{28}$$

If $e^{-4}\theta^2|X| < \theta d|\bar{X}(r)|/2$, then inequalities (28) and (27) imply

$$e^{-4}\theta^2|X| \leq \frac{8\epsilon}{\varepsilon(1-2A\delta)(\delta A)^c}\mathrm{cap}(X),$$

which contradicts the assumption $|X| \geq \lambda\,\mathrm{cap}(X)$ in Property 3.2 for sufficiently small ratio ϵ/ε (depending on c). If $\theta d|\bar{X}(r)|/2 \leq e^{-4}\theta^2|X|$, then inequality (28) immediately implies inequality (20) for sufficiently small ϵ (i.e., sufficiently large d) depending on c.

We have thus proved inequality (20) as well as the first statement of Property 3.2. We next prove the second and third statements of Property 3.2 together.

Let $\eta'\,\mathrm{cap}(X) + |X|/2$ be the number of elements in X whose ranks are at most $m(X)$. By Claim 3.1,

$$\eta' \leq \eta. \tag{29}$$

Without loss of generality, we assume

$$\eta' > 0; \tag{30}$$

i.e., X contains more than $|X|/2$ elements whose ranks are at most $m(X)$. (Note that if we were to prove the second statement only, we could not make such an assumption.)

CLAIM 3.6. *At least* $7\epsilon|X|$ *elements* x*'s of* $S'_{CL}(X)$ *satisfy the inequality*

$$|X|/2 - 4\epsilon|X| \le \mathrm{rank}(x, X) \le |X|/2 + \eta' \mathrm{cap}(X) + (\varepsilon_1 + 16\epsilon)|X|. \tag{31}$$

Proof. For ease of notation, suppose all elements with LARGE values larger than or equal to $d(1/2-2\epsilon)$ are listed according to their LARGE values, with the element of the smallest LARGE values listed first. We call the first at most $\lceil (1/2-\lambda/10)|X| \rceil$ non-empty positions *ideal positions*. (Note that an element of $S_C(X)$ is surely included in $S'_{CR}(X)$ if it is at an ideal position.)

Let

$$IS_{CR}(X) = \{x \in S_C(X) \,|\, \mathrm{rank}(x) \in [m(X), m(X) + (\varepsilon_1 + 10\epsilon)\mathrm{cap}(X)]\}.$$

By the third statement of Claim 3.4, $|IS_{CR}(X)| \ge 10\,\epsilon\,\mathrm{cap}(X)$. Clearly, for all $x \in IS_{CR}(X)$,

$$|X|/2 + \eta' \mathrm{cap}(X) \le \mathrm{rank}(x, X) \le |X|/2 + \eta' \mathrm{cap}(X) + (\varepsilon_1 + 10\,\epsilon)\,\mathrm{cap}(X). \tag{32}$$

Hence, by inequality (30) and the second statement of Lemma 2.3, at most $\epsilon|X|$ elements of $IS_{CR}(X)$ may have their LARGE values less than or equal to $d(1/2 - 2\epsilon)$ at the end of COMPARE(X, $S_C(X)$, G). We call the $9\epsilon|X|$ or more elements of $IS_{CR}(X)$ that have LARGE values greater than or equal to $d(1/2-2\epsilon)$ *ideal elements*. This also shows that there exist at least $9\epsilon|X|$ ideal positions.

Now Claim 3.6 would immediately follow from inequality (32) if at least $9\epsilon|X|$ ideal elements are included in $S'_{CR}(X)$ (the "if" condition will be clearly satisfied when at least $9\epsilon|X|$ ideal elements are in ideal positions). We now prove that although some ideal elements are not necessarily in ideal positions, the ideal positions will contain enough elements satisfying inequality (31).

According to the first statement of Lemma 2.3, at most $\epsilon|X|$ elements whose ranks with respect to X are less than $|X|/2 - 4\epsilon|X|$ may have their LARGE values larger than $d(1/2-2\epsilon)$. This means that at most $\epsilon|X|$ elements violating the first inequality of inequality (31) may occupy at most $\epsilon|X|$ ideal positions. Now, since we have shown that there are at least $9\epsilon|X|$ ideal positions, if the claim is incorrect, then there must exist at least $\epsilon|X|$ elements violating the second inequality of inequality (31) that are located among the first $9\epsilon|X|$ ideal positions, and are therefore located before at least $\epsilon|X|$ ideal elements. By inequalities (31) and (32), this contradicts the third statement of Lemma 2.3. □

Let $\bar{X} = \{x \in FL \cup CL \,|\, \mathrm{rank}(x) \ge m(x)\}$. We next show that

$$|\bar{X}| \le \varepsilon|X|. \tag{33}$$

By the choice of η', all elements in $\bar{X}$ have ranks with respect to X at least $|X|/2 + \eta'\mathrm{cap}(X)$. Thus, by $\epsilon|X| \ge 2$, $\bar{X}$ contains at least $|\bar{X}| - (\varepsilon_1 + 18\epsilon)|X|$ elements whose ranks with respect to X are at least $|X|/2 + \eta'\mathrm{cap}(X) + (\varepsilon_1 + 17\epsilon)|X|$. Therefore, since $\bar{X} \subseteq FL \cup CL$, $\bar{X}$ contains at least $|\bar{X}| - (\varepsilon_1 + 18\epsilon + 6\epsilon)|X|$ elements that have ranks with respect to X at least $|X|/2 + \eta'\mathrm{cap}(X) + (\varepsilon_1 + 17\epsilon)|X|$ but are listed among the first $|X|/2 - 5\epsilon|X|$ positions at the end of PARTITION(X). As a consequence, at least $|\bar{X}| - (\varepsilon_1 + 18\epsilon + 6\epsilon)|X|$ elements whose ranks with respect to X are at most $|X|/2 - 4\epsilon|X|$ are listed after the first $|X|/2 - 5\epsilon|X|$ positions. Therefore, at

least $|\bar{X}| - (\varepsilon_1 + 18\epsilon + 6\epsilon)|X|$ elements whose ranks with respect to X are at least $|X|/2 + \eta' \operatorname{cap}(X) + (\varepsilon_1 + 17\epsilon)|X|$ are listed before at least $|\bar{X}| - (\varepsilon_1 + 18\epsilon + 6\epsilon)|X|$ elements whose ranks with respect to X are at most $|X|/2 - 4\epsilon|X|$. By Claim 3.6 and the third statement of Lemma 2.3, we have

$$|\bar{X}| - (\varepsilon_1 + 18\epsilon + 6\epsilon)|X| \leq \epsilon|X|, \tag{34}$$

which implies inequality (33) for sufficiently small ϵ and ε_1.

Now the third statement of Property 3.2 follows immediately from inequality (33), and the second statement of Property 3.2 follows from

$$|\{x \in FR \cup CR \,|\, \operatorname{rank}(x) \leq m(X)\}| = |\bar{X}| + \eta' \operatorname{cap}(X) \leq \eta \operatorname{cap}(\mathrm{X}) + \varepsilon|X|,$$

where the inequality follows from inequalities (29) and (33). This completes the correctness proof for our entire algorithm.

Analysis of running time. We next prove that our whole algorithm runs in $O(n \log n)$ time. The algorithm proceeds in $O(\log n)$ stages, and so we only need to show that each stage of the algorithm runs in $O(n)$ time. Within each stage, we first partition each AKS tree node X into four parts, FL, CL, CR, and FR, and then we move each of the four parts to the children of X or to the parent of X. Since moving these parts takes constant time for each node, we need only to analyze the time needed to partition X.

Claim 3.7. *The time needed to partition an AKS tree node X is at most*

$$O\left(|S(X)| \log \frac{|S(X)|}{|X|}\right), \tag{35}$$

where $|S(X)|$ denotes the number of elements contained in $S(X)$.

Proof. In the case $\epsilon|X| < 2$, Algorithm Partition(X) makes its comparisons based on a complete bipartite graph between X and $S(X)$. The total number of comparisons is $O(|S(X)|\,|X|) = O(|S(X)|/\epsilon) = O(|S(X)|)$. The construction of the DAG based on these comparison results takes $O(|S(X)|)$ time. Finally, the topological sort of X takes $O(|X|) = O(1)$ time, so the total time needed is $O(|S(X)|)$. The correctness of the claim now follows from inequality (15).

In the rest of the proof for the claim, we assume $\epsilon|X| \geq 2$. In this case, Algorithm Partition(X) consists of four steps. In each of the steps, Algorithm Compare is applied to X and a subset of $S(X)$, and we select a subset of the involved nuts and bolts, based upon their Small and Large values. By inequality (15), the selection of these subsets can be easily done in $O(|S(X)|)$ time by CountingSort. Hence, we only need to analyze the time needed within each execution of Compare.

Algorithm Compare consists of (1) comparisons based on a graph (say, G) of Lemma 2.1 and (2) listing the involved nuts and bolts according to their Small and Large values. Clearly, the listing can be easily done in $O(|S(X)|)$ time by CountingSort. Hence, the total time needed in Compare is determined by the total number of the involved comparisons, which, in turn, is equal to the number of edges in G. By Lemma 2.1, the number of edges in G is at most

$$O\left(|S(X)| \log \frac{|S(X)|}{|X|}\right).$$

This completes the correctness proof of Claim 3.7. ∎

We next give an upper bound on $|S(X)|$. Since $|S(X)|$ and $S_L(X)$ differ by at most a constant factor, we will upper bound $S_L(X)$ instead. By definition, $S_L(X)$ consists of trees of the form $T_{Y'}(\lceil 0.5\,h(X')+c\rceil)$, where $Y' \in P_L(X)$. Since the capacity of an AKS tree node decreases by a factor of A every level upward in the AKS tree, we obtain

$$|S_L(X)| = O\left(\left|T_{X'_{L,1}}(\lceil 0.5\,h(X')+c\rceil)\right|\right), \tag{36}$$

where $|T_{X'_{L,1}}(\lceil 0.5\,h(X')+c\rceil)|$ denotes the number of elements contained in $T_{X'_{L,1}}(\lceil 0.5\,h(X')+c\rceil)$. On the other hand, the total number of elements contained in

$$T_{X'_{L,1}}(\lceil 0.5\,h(X')+c\rceil)$$

is within a constant factor of the total capacity of all the leaves in

$$T_{X'_{L,1}}(\lceil 0.5\,h(X')+c\rceil).$$

Thus, the right-hand side of (36) is at most

$$O\left(2^{h(X'_{L,1})}\,A^{h(X'_{L,1})}\,\mathrm{cap}(X'_{L,1})\right) = O\left(2^{h(X'_{L,1})}\,A^{h(X)}\,\mathrm{cap}(X)\right) = O\left(5^{h(X)}\,\mathrm{cap}(X)\right),$$

where the last equality holds since $h(X'_{L,1}) = \lceil 0.5\,h(X)+c\rceil$. Thus, we have shown

$$|S(X)| = O\left(|S_L(X)|\right) = O\left(5^{h(X)}\,\mathrm{cap}(X)\right). \tag{37}$$

Moreover, in the time analysis of the algorithm, we can assume without loss of generality that

$$|X| \geq \lambda\,\mathrm{cap}(X). \tag{38}$$

Otherwise, CL and CR would be empty and all elements in X would be sent to the parent of X at the end of the stage. In such a case, we would not have to partition X at all. By Claim 3.7 and inequalities (37) and (38), the running time of each stage of the algorithm is at most

$$\begin{aligned}
&O\left(\sum_{X\in T_N}(h(X)+1)\,5^{h(X)}\,\mathrm{cap}(X)\right)\\
&= O\left(\left(\sum_{X\in T_N,\,h(X)=0}\mathrm{cap}(X)\right)\left(\sum_{0\leq h\leq H}(h+1)\left(\frac{5}{6}\right)^h\right)\right)\\
&= O\left(\sum_{X\in T_N,\,h(X)=0}\mathrm{cap}(X)\right)\\
&= O(n),
\end{aligned}$$

where H denotes the height of the whole AKS tree and the first equality follows from

$$\sum_{X\in T_N,\,h(X)=i+1}\mathrm{cap}(X) = \frac{1}{2A}\sum_{X\in T_N,\,h(X)=i}\mathrm{cap}(X).$$

This completes the time analysis of the algorithm. □

COROLLARY 3.2. *When it is allowed to make copies of nuts and bolts, the algorithm can be modified to sort n nuts and n bolts in $O(\log n)$ time on n processors in Valiant's parallel comparison tree model.*

Sketch of Proof. Given the proof of Theorem 3.1, the proof of the corollary is relatively simple. The key fact is that COMPARE(U, V, G) can be executed in a constant number of parallel steps in Valiant's parallel comparison tree model, even if d, the degree of a vertex in V, may not be a constant; we can simply make d copies for each element in V. This modification will not affect the outcome of COMPARE(U, V, G) because within COMPARE(U, V, G) whether an element x should be compared with another element y does not depend on the outcome of any other comparisons that are made earlier during the execution of COMPARE(U, V, G). Moreover, the modification will not increase the total number of comparisons involved in COMPARE(U, V, G). So the total number of processors needed for each of the $O(\log n)$ stages remains linear in n. □

4. Concluding remarks. We have designed an optimal $O(n \log n)$-time algorithm for sorting or matching nuts and bolts. Since our algorithm depends on some random graphs that we do not know how to construct explicitly, a natural open question is how to make our algorithm constructive.

Our algorithm can be executed in optimal $O(\log n)$ time on n processors in Valiant's parallel comparison tree model, provided that we can make copies of nuts and bolts. However, when no copies are allowed (which appears to be a reasonable assumption), we do not know if it is possible to sort the nuts and bolts in $O(\log n)$ time on n processors in Valiant's parallel comparison tree model.

Aumann [5] has pointed out that it is still possible to sort nuts and bolts by a certain algorithm, even if there is no one-to-one match between the nuts and bolts. It is easy to see that, when all different nuts have distinct widths and all different bolts have distinct widths, such sorting is possible if and only if for any pair of nuts (respectively, bolts) there exists a bolt (resp., nut) whose width is between the widths of the pair. It can be shown that our algorithm sorts as long as such sorting is possible. That is, our algorithm sorts distinct nuts and distinct bolts in $O(n \log n)$ optimal sequential time (or $O(\log n)$ optimal parallel time on n processors in Valiant's parallel comparison tree model when copying nuts and bolts is allowed) as long as such sorting is possible by *any* algorithm. Note that when sorting is possible but matching is impossible, even $O(n \log n)$ *expected* sequential time does not seem to be entirely trivial [5].

As we have mentioned in the introduction, the $O(n \log^4 n)$-time algorithm of Alon et al. [3] (resp., the $O(n \log^2 n)$-time algorithm of Bradford and Fleischer [8]) for sorting nuts and bolts is based on an $O(n \log^3 n)$-time (resp., $O(n \log n)$-time) algorithm for selecting a median nut and a median bolt. It is well known that the classic median selection (from a list of n numbers) can be done in $O(n)$ time [6]. It is curious to study if $O(n)$-time median selection is possible in the context of nuts and bolts (say, when there is a match between nuts and bolts). By using the graphs of Lemma 2.1 in some interesting way and by using some technique of [1], we have found an $O(n\,(\log \log n)^2)$-time algorithm for selecting a median nut and a median bolt. This also gives an $O(n \log n\,(\log \log n)^2)$-time algorithm for sorting or matching nuts and bolts. One nice property of this algorithm is that the constant factors behind the "O"

notations are reasonable, as opposed to the prohibitively large constant involved in our AKS-based approach. Details of our median-selection algorithm are omitted. We recently heard that Bradford [7] found an $O(n)$-time algorithm for finding a median nut and a median bolt, which apparently yields another $O(n \log n)$-time sequential algorithm for sorting or matching nuts and bolts.

Acknowledgments. We thank Noga Alon for telling us the problem before [3] was published, Greg Plaxton for stimulating discussion on the design of the partition scheme described in subsection 2.2.3, and Yonatan Aumann, Nabil Kahale, and Tom Leighton for helpful conversations.

REFERENCES

[1] M. Ajtai, J. Komlós, W. L. Steiger, and E. Szemerédi, *Optimal parallel selection has complexity* $O(\log \log N)$, J. Comput. System Sci., 38 (1989), pp. 125–133.

[2] M. Ajtai, J. Komlós, and E. Szemerédi, *Sorting in* $c \log n$ *parallel steps*, Combinatorica, 3 (1983), pp. 1–19.

[3] N. Alon, M. Blum, A. Fiat, S. Kannan, M. Naor, and R. Ostrovsky, *Matching nuts and bolts*, in Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms, Arlington, VA, 1994, pp. 690–696.

[4] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley-Interscience, New York, 1991.

[5] Y. Aumann, private communication, MZT, 1994.

[6] M. Blum, R. Floyd, V. Pratt, R. Rivest, and R. Tarjan, *Time bounds for selection*, J. Comput. System Sci., 7 (1973), pp. 448–461.

[7] P. G. Bradford, private communication, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1995.

[8] P. G. Bradford and R. Fleischer, *Matching nuts and bolts faster*, in Proc. 6th Internat. Symp. Algorithms and Computation, Cairns, Australia, Lecture Notes in Comput. Sci., Springer-Verlag, 1995, pp. 402–408.

[9] M. S. Paterson, *Improved sorting networks with* $O(\log N)$ *depth*, Algorithmica, 5 (1990), pp. 75–92.

[10] G. J. E. Rawlins, *Compared to What? An Introduction to the Analysis of Algorithms*, Computer Science Press, Rockville, MD, 1991.

[11] L. G. Valiant, *Parallelism in comparison problems*, SIAM J. Comput., 4 (1975), pp. 348–355.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 373–380, August 1998

002

A BOUND OF 4 FOR THE DIAMETER OF THE SYMMETRIC TRAVELING SALESMAN POLYTOPE*

FRED J. RISPOLI† AND STEVEN COSARES†

Abstract. We investigate the diameter of the polytope arising in the n-city symmetric traveling salesman problem (TSP) and perfect matching polytopes. Grötschel and Padberg [*The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley-Intersci. Ser. Discrete Math., E. Lawler et al., eds., John Wiley, Chichester, 1985, pp. 251–305] conjectured that the diameter of the symmetric TSP polytope is 2, independent of n. We constructively show that its diameter is at most 4, for all $n \geq 3$. Our result also shows that the diameter of the perfect 2-matching polytope is at most 6, for every $n \geq 3$.

Key words. diameter, polytope, travelling salesman problem

AMS subject classification. 90C35

PII. S0895480196312462

1. Introduction.

The *symmetric traveling salesman problem* (TSP) associated with the complete graph K_n is the problem of finding a tour having the smallest possible total distance, where a distance is assigned to every edge of K_n. The *symmetric traveling salesman polytope* associated with K_n is TSP_n = convex hull $\{x^T\colon T$ is a tour of $K_n\}$, where x^T is the incidence vector associated to the edges in T. The polyhedral structure of TSP_n has been studied by many. In [2] Grötschel and Padberg conjectured that the diameter of TSP_n is 2. Investigations by Sierksma and Tijssen resulted in a series of papers, and the following upper bounds were successively obtained: $n-2$, $n-\lfloor\sqrt{n-2}\rfloor$, and $\lfloor n/2\rfloor$ (see [7]). In this paper, a constructive proof is given showing that the diameter of TSP_n is at most 4, for every $n \geq 3$, and is thus independent of n.

Papadimitriou [5] showed that the problem of determining if two extreme points of TSP_n are nonadjacent is NP-complete. Nevertheless, there exists a characterization of neighboring extreme points for the perfect 2-matching polytope that can be exploited to provide a sufficient condition for adjacency on TSP_n (see Rispoli [6]). Padberg and Rao used a similar approach to adjacency on the asymmetric traveling salesman polytope to show that its diameter is 2 (see [4]). They also determined that the diameter of the perfect matching polytope is 2. Here we give a new proof of this fact. The technique we use will also be employed to obtain the bound for TSP_n. An intermediate result is that for every pair of tours having a perfect matching in common, the distance between their corresponding extreme points is at most 2. We also provide the first constant upper bound on the diameter of the perfect 2-matching polytope associated with K_n. In particular, we show that its diameter is at most 6.

The outline of this paper is as follows. Basic definitions are given in the next section, along with a discussion of the perfect matching polytope. Next, we obtain the bound for the diameter of TSP_n. The paper ends with some concluding remarks and a discussion of the perfect 2-matching polytope.

*Received by the editors November 25, 1996; accepted for publication (in revised form) December 15, 1997; published electronically July 7, 1998.
http://www.siam.org/journals/sidma/11-3/31246.html
†Department of Mathematics, Dowling College, Oakdale, NY 11769 (rispolif@dowling.edu).

2. Preliminaries and the perfect matching polytope. The *distance* between a pair of extreme points of a polytope P is the number of extreme edges in a shortest path linking them. The *diameter* of P, $\delta(P)$, is the largest of the distances between pairs of extreme points of P. The study of the function δ is motivated by its relationship to edge-following algorithms of linear programming such as the simplex method. For a comprehensive survey see Klee and Kleinschmidt [3].

For any two sets S and T, let $S \Delta T$ denote the *symmetric difference* $(S \sim T) \cup (T \sim S)$ or equivalently $(S \cup T) \sim (S \cap T)$. Let K_n be the complete undirected graph on nodes $N = \{1, 2, \ldots, n\}$ with edges $\{\{i, j\} : i, j \in N \text{ and } i \neq j\}$. A subgraph H of K_n is called *acyclic* if there are no cycles in H. Given two subsets of edges H_1 and H_2, an *alternating cycle* is a cycle of even length whose edges alternate between H_1 and H_2. A *simple alternating cycle* is an alternating cycle whose edges cannot be partitioned into two or more edge-disjoint alternating cycles. A *matching* is a set of edges in K_n such that no two edges have a node in common. For every even $n \geq 2$, a *perfect matching* is a matching of K_n consisting of $n/2$ edges. For every $n \geq 3$, a *perfect 2-matching* is a subgraph of K_n consisting of n edges and all nodes of degree 2. A *tour* T is a subgraph of K_n that is a perfect 2-matching and is also a cycle of length n. Notice that for every even $n \geq 4$, T can be partitioned into two subgraphs M_1 and M_2 which are both perfect matchings. For every odd $n \geq 3$, $T = M_1 \cup M_2$, where M_1 is a matching with $(n-1)/2$ edges, M_2 is a subgraph containg $(n+1)/2$ edges, and $M_1 \cap M_2 = \oslash$.

LEMMA 1. *Let $n \geq 4$ be even. Let M_1 and M_2 be perfect matchings in K_n. Then there exists a perfect matching M in K_n such that $M_1 \cup M$ and $M_2 \cup M$ are both tours of K_n.*

Proof. The proof is by induction on n. The case $n = 4$ can be checked directly. Assume that the statement holds for some even $n \geq 4$.

Suppose M_1 and M_2 are perfect matchings of K_{n+2}. Clearly, there is an edge in K_{n+2} that is not in $M_1 \cup M_2$, say $e = \{u, v\}$. $M_1 \cup \{e\}$ contains exactly one connected component with three edges of the form $\{g, u\} \cup \{u, v\} \cup \{h, v\}$, with $g \neq h$. Let H_1 be the perfect matching in K_n obtained from M_1 by removing nodes u and v and replacing the 3-edge component with edge $\{g, h\}$. Obtain H_2 by similarly removing u and v and "contracting" the 3-edge component in $M_2 \cup \{e\}$. By the inductive assumption, there is a perfect matching H in K_n such that $H_1 \cup H$ and $H_2 \cup H$ are both tours in K_n. Now, $M = H \cup \{e\}$ satisfies the lemma. □

The *perfect matching polytope* associated with K_n, for n even, is PM_n = convex hull $\{x^M : M \text{ is a perfect matching of } K_n\}$, where x^M is the incidence vector associated to the edges in M. The perfect matchings in K_n are in one-to-one correspondence with the extreme points of PM_n. It is well known that if M_1 and M_2 are perfect matchings in K_n, then the extreme points corresponding to M_1 and M_2 are neighbors if and only if $M_1 \Delta M_2$ contains a unique simple alternating cycle (see [1]). Lemma 1 provides a concise proof of the following theorem; the original proof is in [4].

THEOREM 1 (Padberg and Rao). *$\delta(PM_n) = 2$, for every even $n \geq 8$, and $\delta(PM_n) = 1$, for $n = 4$ and 6.*

Proof. For every pair of extreme points x^{M_1} and x^{M_2}, Lemma 1 implies the existence of an extreme point x^M adjacent to both x^{M_1} and x^{M_2}, hence $\delta(PM_n) \leq 2$. It is easy to find a pair of extreme points requiring exactly two steps. □

The *perfect 2-matching polytope* associated with K_n, for $n \geq 3$, is PTM_n = convex hull $\{x^M : M \text{ is a perfect 2-matching of } K_n\}$, where x^M is the incidence vector associated with the edges in M. Since every tour in K_n is a perfect 2-matching, and

a complete nonredundant system of equations and inequalities describing PTM_n is known, the perfect 2-matching polytope is helpful in determining facets of TSP_n (see [2] for more details). The perfect 2-matchings of K_n are in one-to-one correspondence with the extreme points of PTM_n, and, likewise, the tours of K_n are in one-to-one correspondence with the extreme points of TSP_n. Two perfect 2-matchings are called *adjacent* if their corresponding extreme points are adjacent on PTM_n. Similarly, we call tours adjacent if their corresponding extreme points are adjacent on TSP_n. A proof of Lemma 2 is provided in [6], where the "monotonic diameter" of PTM_n is obtained.

LEMMA 2. *If M_1 and M_2 are a pair of perfect 2-matchings in K_n, then M_1 and M_2 are adjacent on PTM_n if and only if $M_1 \Delta M_2$ contains a unique simple alternating cycle.*

3. The symmetric traveling salesman polytope. Since every tour is a perfect 2-matching, TSP_n is contained in PTM_n. But what can be said about neighboring extreme points? If two tours are adjacent on TSP_n, they may not be adjacent on PTM_n. However, if two tours are adjacent on PTM_n, then they must necessarily be adjacent on TSP_n. By Lemma 2, we know that if T_1 and T_2 are tours and $T_1 \Delta T_2$ contains a unique simple alternating cycle, then T_1 and T_2 are adjacent on PTM_n. This gives the following.

LEMMA 3. *Let T_1 and T_2 be tours of K_n. If $T_1 \Delta T_2$ contains a unique simple alternating cycle, then T_1 and T_2 are adjacent on TSP_n.*

The following notation will now be helpful. Given a subgraph H of K_n consisting of q disjoint paths $\rho_1, \rho_2, \ldots, \rho_q$ of length one or more, let $N(H)$ denote the subset of nodes consisting of the $2q$ endpoints of the ρ_i. Let $PM(H)$ denote the perfect matching on $N(H)$ induced by H by representing each ρ_i in H with an edge e_i having the same endpoints as ρ_i, for every $i = 1, \ldots, q$. In other words, $PM(H)$ consists of the graph obtained from H by contracting each path ρ_i into an edge e_i having the same endpoints as ρ_i.

THEOREM 2. (a) *Let $n \geq 4$ be even. Then for every pair of tours T_1 and T_2 of K_n having a perfect matching in common, the distance between their corresponding extreme points is at most 2 on* TSP_n.

(b) *Let $n \geq 5$ be odd. Then for every pair of tours T_1 and T_2 of K_n having a matching with $(n-1)/2$ edges in common, the distance between their corresponding extreme points is at most 2 on TSP_n.*

Proof. (a) The case $n = 4$ may be checked directly, so assume that $n \geq 6$. Let T_1 and T_2 be nonadjacent tours of K_n satisfying $T_1 = M_1 \cup M_2$ and $T_2 = M_1 \cup M_3$, where M_1, M_2, and M_3 are perfect matchings in K_n. Since $T_1 \Delta T_2 \subseteq M_2 \cup M_3$, and T_1 and T_2 are not adjacent, $M_2 \cup M_3$ consists of at least two components which are either alternating cycles of length four or more, or single edges in $M_2 \cap M_3$. Let $C_1, \ldots, C_q$, denote the components in $M_2 \cup M_3$. Let g_i be any edge in $C_i \cap M_3$, for every $i = 1, \ldots, q$. Set $G = \{g_1, \ldots, g_q\}$ and set $E = M_3 \sim G$. Observe that $M_1 \cup E$ must be acyclic since it is contained in T_2. Moreover, $PM(M_1 \cup E)$ and $PM(M_2 \cup E)$ are perfect matchings on $N(M_1 \cup E)$. By Lemma 1, there is a perfect matching H that links both $PM(M_1 \cup E)$ and $PM(M_2 \cup E)$ into a tour of $N(M_1 \cup E)$. Let $T_3 = M_1 \cup E \cup H$; then T_3 is a tour of K_n and satisfies

$$T_1 \Delta T_3 = (M_1 \cup M_2) \Delta (M_1 \cup E \cup H) = M_2 \cup E \cup H$$

which is also a tour of K_n. So, by Lemma 3, T_1 and T_3 are adjacent. In addition, by

the definition of E, $M_3 \sim E$ is the same as $PM(M_2 \cup E)$, and hence

$$T_2 \,\Delta\, T_3 = (M_1 \cup M_3) \,\Delta\, (M_1 \cup E \cup H) = (M_3 \sim E) \cup H$$

is a simple alternating cycle. Therefore, T_2 and T_3 are adjacent.

(b) The case $n = 5$ may be checked directly, so assume that $n \geq 7$. Let T_1 and T_2 be nonadjacent tours of K_n having a matching with $(n-1)/2$ edges in common. Without loss of generality, we can assume that $T_1 = M_1 \cup M_2$ and $T_2 = M_1 \cup M_3$, where M_1 is a perfect matching on $N \sim \{1\}$, $M_1 \cap M_2 = \oslash$, and $M_1 \cap M_3 = \oslash$. Observe that either (i) the degree of node 1 is 0 or 2 in $T_1 \,\Delta\, T_2$; or (ii) the degree of node 1 is 4 in $T_1 \,\Delta\, T_2$ and node 1 is incident to one simple alternating cycle in $T_1 \,\Delta\, T_2$; or (iii) the degree of node 1 is 4 in $T_1 \,\Delta\, T_2$ and node 1 is incident to two simple alternating cycles in $T_1 \,\Delta\, T_2$.

(i) Let f_1 be an edge in $M_2 \cap M_3$ incident to node 1, and let f_2 be the other edge in M_3 incident to node 1, which may or may not be in $M_2 \cap M_3$. Let $C_1, \ldots, C_q$ denote the components in $M_2 \cup M_3$ that do not contain node 1. Let g_i be any edge in $C_i \cap M_3$, for every $i = 1, \ldots, q$. Set $G = \{f_2, g_1, \ldots, g_q\}$ and set $E = M_3 \sim G$. Observe that both $M_1 \cup E$ and $M_2 \,\Delta\, E$ must be acyclic. Moreover, $PM(M_1 \cup E)$ and $PM(M_2 \,\Delta\, E)$ are perfect matchings on $N(M_1 \cup E)$. By Lemma 1, there is a perfect matching H on $N(M_1 \cup E)$ linking both $PM(M_1 \cup E)$ and $PM(M_2 \,\Delta\, E)$ into a tour of $N(M_1 \cup E)$. Hence $T_3 = M_1 \cup E \cup H$ is a tour of K_n, and T_1 and T_3 are adjacent since

$$T_1 \,\Delta T_3 = (M_1 \cup M_2) \,\Delta\, (M_1 \cup E \cup H) = M_2 \,\Delta\, (E \cup H) = (M_2 \,\Delta\, E) \cup H$$

is a simple alternating cycle. In addition, $M_3 \sim E = PM(M_2 \,\Delta\, E)$, hence

$$T_2 \,\Delta\, T_3 = (M_1 \cup M_3) \,\Delta\, (M_1 \cup E \cup H) = M_3 \,\Delta\, (E \cup H) = (M_3 \sim E) \cup H$$

is also a simple alternating cycle, implying that T_1 and T_3 are adjacent.

(ii) Let γ denote the simple alternating cycle containing node 1 in $T_1 \,\Delta\, T_2$. Let f be any edge in $\gamma \cap M_3$ that is not incident to node 1. Let $C_1, \ldots, C_q$ denote the components in $M_2 \cup M_3$ that do not contain node 1. Let g_i be any edge in $C_i \cap M_3$, for every $i = 1, \ldots, q$. Set $G = \{f, g_1, \ldots, g_q\}$ and $E = M_3 \sim G$. Then $M_1 \cup E$ is acyclic, and $M_2 \cup E$ consists of disjoint alternating paths save that component containing node 1 which consists of $\gamma \sim \{f\}$. Moreover, $PM(M_1 \cup E)$ and $PM[(M_2 \cup E) \sim \gamma] \cup \{f\}$ are perfect matchings on $N(M_1 \cup E)$. By Lemma 1, there is a perfect matching H on $N(M_1 \cup E)$ linking both $PM(M_1 \cup E)$ and $PM[(M_2 \cup E) \sim \gamma] \cup \{f\}$ into a tour of $N(M_1 \cup E)$. Hence $T_3 = M_1 \cup E \cup H$ is a tour of K_n, and T_1 and T_3 are adjacent since

$$T_1 \,\Delta\, T_3 = (M_1 \cup M_2) \Delta (M_1 \cup E \cup H) = M_2 \cup E \cup H$$

is a simple alternating cycle. In addition, $M_3 \sim E = PM[(M_2 \cup E) \sim \gamma] \cup \{f\}$, so

$$T_2 \,\Delta\, T_3 = (M_1 \cup M_3) \,\Delta\, (M_1 \cup E \cup H) = M_3 \,\Delta\, (E \cup H) = (M_3 \sim E) \cup H$$

is also a simple alternating cycle. Thus, T_2 and T_3 are adjacent.

(iii) Now, node 1 is incident to exactly two simple alternating cycles in $T_1 \,\Delta\, T_2$, say, γ_1 and γ_2. Let f_1 be any edge in $\gamma_1 \cap M_3$ that is not incident to node 1, and let $f_2 = \{1, k\}$ be the unique edge in $\gamma_2 \cap M_3$ incident to node 1. Let $C_1, \ldots, C_q$ denote the components in $M_2 \cup M_3$ that do not contain node 1. Let g_i be any edge

in $C_i \cap M_3$, for every $i = 1, \ldots, q$. Set $G = \{f_1, f_2, g_1, \ldots, g_q\}$ and $E = M_3 \sim G$. Then $M_1 \cup E$ is acyclic and consists of disjoint alternating paths, where all paths begin and terminate with M_1 edges except for the path having node 1 as an endpoint which begins with an M_3 edge and terminates with an M_1 edge. In addition, $M_2 \cup E$ consists of disjoint alternating paths save that component containing node 1 which consists of $(\gamma_1 \cup \gamma_2) \sim \{f_1, f_2\}$.

Set $H_1 = PM(M_1 \cup E)$ and $H_2 = PM[(M_2 \cup E) \sim (\gamma_1 \cup \gamma_2)] \cup \{f_1, f_2\}$. Then H_1 and H_2 are perfect matchings on $N(M_1 \cup E)$. If $T_1 \,\Delta\, T_2 = \gamma_1 \cup \gamma_2$, then either T_1 and T_2 are neighbors or there exists a T_3 obtained by exchanging M_2 and M_3 edges along any one of γ_1 or γ_2. So we may assume that $T_1 \,\Delta\, T_2$ contains at least three simple alternating cycles, implying that $N(M_1 \cup E)$ contains at least six nodes, four of which are the endpoints of f_1 and f_2.

Next we define a procedure to find a subset of edges H linking $M_1 \cup E$ into a tour of K_n and $M_2 \cup E$ into a subgraph containing a unique alternating cycle. Initially set $\underline{H}_1 = H_1$ and $\underline{H}_2 = H_2$. Let $\{1, v_1\}$ be the edge in H_1 incident to node 1, and let $\{v_2, k\}$ be the edge in H_1 incident to node k. $H_1 \cup H_2$ must contain an alternating cycle of length 4 or more passing through node 1. If this cycle has length exactly 4, then set $e = \{1, v_2\}$, place edge $\{v_1, k\}$ in H, and contract $\underline{H}_1$ and $\underline{H}_2$ with respect to $\{v_1, k\}$. Otherwise, the cycle in $H_1 \cup H_2$ passing through node 1 has length at least 6, and hence edge $\{v_1, v_2\}$ is not in H_2. Now set $e = \{1, k\}$, place edge $\{v_1, v_2\}$ in H, and contract $\underline{H}_1$ and $\underline{H}_2$ with respect to $\{v_1, v_2\}$. Observe that in every case, e is incident to node 1 and, after the contraction, $e \in \underline{H}_1 \cap \underline{H}_2$. Next, iteratively choose an edge $\{g, h\}$ that joins two nodes in $N(\underline{H}_1 \sim \{e\})$ satisfying $\{g, h\} \notin \underline{H}_1 \cup \underline{H}_2$. Place $\{g, h\}$ in H, and contract $\underline{H}_1$ and $\underline{H}_2$ with respect to $\{g, h\}$. This gives new perfect matchings on a smaller subset of nodes denoted by $\underline{H}_1$ and $\underline{H}_2$ throughout this procedure which is repeated until both $\underline{H}_1$ and $\underline{H}_2$ contain two edges. Upon completion, $\underline{H}_1 = \underline{H}_2 = \{e, \{u, v\}\}$, for some edge $\{u, v\}$ that represents a unique path in $M_2 \cup E \cup H$, say ρ, having odd length and passing through node 1. Moreover, either the distance on ρ from node 1 to u is odd, or the distance on ρ from 1 to v is odd. In the first case, complete the construction of H by adding to H edge $\{1, v\}$, and add either $\{k, u\}$, if $e = \{1, k\}$, or $\{u, v_2\}$, if $e = \{1, v_2\}$. When the distance on ρ from 1 to v is odd, add to H edge $\{1, u\}$, and add either $\{k, v\}$, if $e = \{1, k\}$, or $\{v_2, v\}$ if $e = \{1, v_2\}$. Notice that $M_2 \cup E \cup H$ now consists of a single alternating cycle that can be partitioned into two odd cycles meeting only at node 1. Finally, set $T_3 = M_1 \cup E \cup H$. Then T_3 is a tour of K_n, and T_3 is adjacent to both T_1 and T_2 since

$$T_1 \,\Delta\, T_3 = (M_1 \cup M_2) \,\Delta\, (M_1 \cup E \cup H) = (M_2 \cup E) \cup H, \text{ and}$$
$$T_2 \,\Delta\, T_3 = (M_1 \cup M_3) \,\Delta\, (M_1 \cup E \cup H) = (M_3 \sim E) \cup H. \qquad \square$$

At this point we remark that $\delta(TSP_n) \leq 6$, for every even $n \geq 4$. For example, suppose that $T_1 = M_1 \cup M_2$ and $T_2 = M_3 \cup M_4$ are arbitrary tours and each M_i is a perfect matching. By Lemma 1, there is a perfect matching M such that $M \cup M_2$ and $M \cup M_3$ are also tours. Now apply Theorem 2(a) three times to link T_1 to $M \cup M_2$, $M \cup M_2$ to $M \cup M_3$, and $M \cup M_3$ to T_2.

LEMMA 4. *Let $n \geq 8$ be even. Let M_1, M_2, M_3, and M_4 be perfect matchings in K_n such that $M_1 \cup M_2$ and $M_3 \cup M_4$ are tours of K_n, and neither $M_2 \cup M_3$ nor $M_2 \cup M_4$ are tours of K_n. Then there exists a perfect matching M in K_n such that $M \cup M_1$, $M \cup M_3$, and $M \cup M_4$ are all tours of K_n, and $M \,\Delta\, M_2$ contains a unique alternating cycle.*

Proof. Let E be a maximal subset of edges in $M_2 \sim (M_3 \cup M_4)$ such that both $M_3 \cup E$ and $M_4 \cup E$ are acyclic. Note that $E \neq \oslash$. Otherwise, $M_2 \sim (M_3 \cup M_4) = \oslash$, implying that either $M_2 = M_3$ or $M_2 = M_4$, contrary to the assumptions. Begin the construction of M by placing every edge in E in M. Set $H_i = PM(M_i \cup E)$, for $i = 1$, 3 and 4, and set $H_2 = M_2 \sim E$. Note that every H_i is a perfect matching on $N(H_1)$. Moreover, $H_2 \subset (H_3 \cup H_4)$. This follows from the fact that if $\{g, h\} \in H_2$ and is not an edge in $H_3 \cup H_4$, then $H_k \cup \{g, h\}$, and hence $H_k \cup E \cup \{g, h\}$ is acyclic for $k = 3$ and 4. This contradicts the maximality of E.

If $H_3 = H_4$, then $H_2 = H_3 = H_4$. By Lemma 1, there is a perfect matching H on $N(H_1)$ such that $H_1 \cup H$ and $H_2 \cup H$ are both tours of $N(H_1)$. Setting $M = E \cup H$ gives the result for this case. So assume that $H_3 \neq H_4$ and that $H_3 \cup H_4$ has at least one cycle of length 4 or more. In addition, $H_3 \cup H_4$ must have more than one component. Otherwise, $H_2 \subset (H_3 \cup H_4)$ implies that either $H_2 = H_3$ or $H_2 = H_4$, say, $H_2 = H_3$. Then $H_2 \cup H_4$, and hence $(M_2 \sim E) \cup (M_4 \cup E)$ is a single component, i.e., a tour, which is impossible. So we may assume that $H_3 \cup H_4$ has at least two components.

Let $N_1 \cup N_2$ be a partition of $N(H_1)$, such that N_1 contains all of the nodes of one of the components of $H_3 \cup H_4$, and N_2 contains all of the nodes from all of the remaining components of $H_3 \cup H_4$. Since at least one of the components in $H_3 \cup H_4$ is a cycle of length 4 or more, we can assume that $|N_1| \geq 4$ and $|N_2| \geq 2$. In addition, $H_1 \cup H_2$ a tour of $N(H_1)$, and $H_2 \subset (H_3 \cup H_4)$ imply that there exists an edge $e = \{v_1, v_2\}$ satisfying $v_1 \in N_1$, $v_2 \in N_2$, and $e \in H_1$. Next we describe a procedure to complete the construction of M. Begin by setting $\underline{H}_i = H_i$, for $i = 1$, 2, 3 and 4, and $\underline{N}_i = N_i$, for $i = 1$ and 2. At each iteration throughout this procedure an edge will be selected, placed in M, and all of the $\underline{H}_i$ will be contracted with respect to this edge. We will continue to refer to the four sequences of perfect matchings obtained as $\underline{H}_i$. Similarly, at every iteration two nodes will be removed from either $\underline{N}_1$ or $\underline{N}_2$, and we continue to refer to the two sequences of subsets of nodes as $\underline{N}_i$. The following step is repeated until it is no longer possible:

> Let $\{g, h\}$ be any edge that is not in $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ such that $\{g, h\}$ joins two nodes in the same $\underline{N}_i$, and neither g nor h are endpoints of e. Place $\{g, h\}$ in M, contract all $\underline{H}_i$ with respect to $\{g, h\}$, and remove g and h from the appropriate $\underline{N}_i$.

When $|\underline{N}_i| \geq 6$, there is always an edge $\{g, h\}$ available. Suppose that $|\underline{N}_i| = q \geq 6$, and consider the complete graph K_q. Observe that q must be even, there are at most $3(q/2) - 1$ edges linking two nodes within the same $\underline{N}_i$, and there are at most $q - 3$ additional edges incident to e in K_q. Since K_q has $q(q-1)/2$ edges, the number of edges available to choose $\{g, h\}$ from is at least

$$[q(q-1)/2] - [3(q/2) - 1] - (q - 3) = (1/2)(q-4)(q-2) > 0, \quad \text{for } q \geq 6.$$

Therefore, upon completion of the above step either $|\underline{N}_1| = 4$ and there are five edges in $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ joining a pair of nodes in $\underline{N}_1$, or $|\underline{N}_1| = 2$. The same is true for $\underline{N}_2$. Thus, $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ now has one of the following forms:

(i) $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ has eight nodes, $|\underline{N}_1| = 4$, and $|\underline{N}_2| = 4$; or
(ii) $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ has six nodes, $|\underline{N}_1| = 4$, and $|\underline{N}_2| = 2$; or
(iii) $\underline{H}_1 \cup \underline{H}_3 \cup \underline{H}_4$ has four nodes, $|\underline{N}_1| = 2$, and $|\underline{N}_2| = 2$.

(i) We know that $v_1 \in \underline{N}_1$, $v_2 \in \underline{N}_2$, and $e \in \underline{H}_1$. Since $\underline{N}_1 \sim \{v_1\}$ has three nodes, there must be an edge, say, $\{v_3, v_4\}$ in $\underline{H}_1$, satisfying $v_3 \in \underline{N}_1$ and $v_4 \in \underline{N}_2$. Let v_5 and v_6 denote the remaining nodes in $\underline{N}_1$, and let v_7 and v_8 denote the remaining

nodes in $\underline{N}_2$. $\{v_5, v_6\}$ must be an edge in $\underline{H}_1$; otherwise the repeated step above would not terminate with $|\underline{N}_1| = 4$. Likewise, $\{v_7, v_8\}$ is in $\underline{H}_1$. In addition, $|\underline{N}_1| = 4$ and $|\underline{N}_2| = 4$ implies that $\underline{H}_1$, $\underline{H}_3$, and $\underline{H}_4$ are mutually disjoint. Suppose that

$$\underline{H}_3 = \{\{v_1, v_5\}, \{v_3, v_6\}, \{v_2, v_7\}, \{v_4, v_8\}\} \quad \text{and}$$
$$\underline{H}_4 = \{\{v_1, v_6\}, \{v_3, v_5\}, \{v_2, v_8\}, \{v_4, v_7\}\}.$$

Then the construction of M is completed by placing $\{\{v_1, v_8\}, \{v_3, v_7\}, \{v_2, v_5\}, \{v_4, v_6\}\}$ in M. There are three other possible combinations for $\underline{H}_3$ and $\underline{H}_4$ but they are all isomorphic to this case.

(ii) Again we know that $v_1 \in \underline{N}_1$, $v_2 \in \underline{N}_2$, $e \in \underline{H}_1$, and there must be an edge, say $\{v_3, v_4\}$ in $\underline{H}_1$, satisfying $v_3 \in \underline{N}_1$ and $v_4 \in \underline{N}_2$. There must also be an edge $\{v_5, v_6\}$ in $\underline{H}_1$ satisfying v_5 and $v_6 \in \underline{N}_1$. Notice that $\{v_2, v_4\}$ is an edge in both $\underline{H}_3$ and $\underline{H}_4$. Since there must be five edges joining a pair of nodes in $\underline{N}_1$, $\underline{H}_3$ and $\underline{H}_4$ have exactly one edge in common. So we can assume that $\underline{H}_3 = \{\{v_1, v_5\}, \{v_2, v_4\}, \{v_3, v_6\}\}$ and $\underline{H}_4 = \{\{v_1, v_6\}\ \{v_2, v_4\}, \{v_3, v_5\}\}$. The construction of M is completed by placing $\{\{v_1, v_3\}, \{v_2, v_5\}, \{v_4, v_6\}\}$ in M.

(iii) Again we know that $v_1 \in \underline{N}_1$, $v_2 \in \underline{N}_2$, $e \in \underline{H}_1$, and that there must be an edge, say $\{v_3, v_4\}$ in $\underline{H}_1$, satisfying $v_3 \in \underline{N}_1$ and $v_4 \in \underline{N}_2$. Moreover, $\underline{H}_3 = \underline{H}_4 = \{\{v_1, v_3\}, \{v_2, v_4\}\}$. The construction of M is completed by placing $\{\{v_1, v_4\}, \{v_2, v_3\}\}$ in M. □

THEOREM 3. *$\delta(TSP_n) \leq 4$, for every $n \geq 3$.*

Proof. For $3 \leq n \leq 7$, the result may be checked directly. Therefore, suppose that $n \geq 8$ and let T and T^* be nonadjacent tours of K_n. We show how to construct a sequence of tours of K_n, denoted by $T = T_0$, T_1, T_2, T_3, $T_4 = T^*$, such that two successive T_i are either identical or adjacent.

Suppose n is even. Then $T_0 = M_1 \cup M_2$ and $T_4 = M_3 \cup M_4$, where the M_i are perfect matchings in K_n. If $M_1 \cup M_3$ is a tour, then set $T_2 = M_1 \cup M_3$. If T_0 and T_2 are adjacent, then set $T_1 = T_2$; otherwise, use Theorem 2(a) to obtain T_1. Obtaining T_3 is similar. Furthermore, the case where any of $M_1 \cup M_4$, $M_2 \cup M_3$, and $M_2 \cup M_4$ are tours is similar. So assume that none of these are tours. By Lemma 4, there is a perfect matching M in K_n such that $M \cup M_1$ and $M \cup M_3$ are both tours of K_n, and $M \,\Delta\, M_2$ and $M \,\Delta\, M_4$ contain unique simple alternating cycles. Set $T_1 = M \cup M_1$ and $T_3 = M \cup M_3$. Then $T_0 \,\Delta\, T_1 = M \,\Delta\, M_2$, so T_0 and T_1 are adjacent. Similarly, T_3 and T_4 are adjacent. By Theorem 2(a), either T_1 and T_3 are adjacent or there is a tour T_2 adjacent to both T_1 and T_3.

Suppose n is odd. Then $T_0 = M_1 \cup M_2$ and $T_4 = M_3 \cup M_4$, where M_2 and M_4 are perfect matchings on $N \sim \{1\}$, $M_1 \cap M_2 = \oslash$, and $M_3 \cap M_4 = \oslash$. If $M_2 \cup M_3$ is a tour of K_n, then set $T_2 = M_2 \cup M_3$. If T_0 and T_2 are adjacent, then set $T_1 = T_2$. Otherwise, M_2 is a matching with $(n-1)/2$ edges, so by Theorem 2(b), there is a tour T_1 adjacent to T_0 and T_2. Moreover, M_3 contains a matching with $(n-1)/2$ edges, so the distance between T_2 and T_4 is at most 2 as well. If $M_2 \cup M_3$ is not a tour of K_n, and $M_2 \cup M_4$ is a tour of $N \sim \{1\}$, then let $C_1, \ldots, C_q$ denote the components in $M_2 \cup M_3$. Since $M_2 \cup M_3$ contains at least two components, $q \geq 2$. Let g_i be any edge in $C_i \cap M_3$, for every $i = 1, \ldots, q$. Set $G = \{g_1, \ldots, g_q\}$ and set $E = M_3 \sim G$. Observe that $M_4 \cup E$ must be acyclic since it is contained in T_4. Moreover, $PM(M_2 \cup E)$ and $PM(M_4 \cup E)$ are perfect matchings on $N(M_2 \cup E)$. By Lemma 1, there is a perfect matching H that links both $PM(M_2 \cup E)$ and $PM(M_4 \cup E)$ into a tour of $N(M_2 \cup E)$. Let $T_2 = M_2 \cup E \cup H$. Then T_2 is a tour of K_n, and T_0 and T_2 have M_2 in common. By Theorem 2(b), the distance between T_0 and T_2 is at most 2. Set

$T_3 = M_4 \cup E \cup H$. Then $T_2 \,\Delta\, T_3 = (M_2 \cup M_4)$, so T_2 and T_3 are adjacent. Moreover, $T_3 \,\Delta\, T_4 = (M_4 \cup E \cup H) \,\Delta\, (M_3 \cup M_4) = (M_3 \sim E) \cup H$. Since $M_3 \sim E$ is the same as $PM(M_2 \cup E)$, $T_3 \,\Delta\, T_4$ is a simple alternating cycle, and hence T_3 and T_4 are adjacent.

Now we may assume that $M_2 \cup M_3$ is not a tour of K_n and $M_2 \cup M_4$ is not a tour of $N \sim \{1\}$. By contracting the 2-edge component in M_1 and M_3 into a single edge and deleting node 1, it follows from Lemma 4 that there is a perfect matching M on nodes $N \sim \{1\}$ such that $M \cup M_1$ and $M \cup M_3$ are both tours of K_n, and $M \,\Delta\, M_2$ and $M \,\Delta\, M_4$ contain unique alternating cycles. Let $T_1 = M \cup M_1$ and $T_3 = M \cup M_3$. Then the distance between T_1 and T_3 is at most 2; hence there exists a path of length at most 4 joining T_0 to T_4. □

4. Concluding remarks and the perfect 2-matching polytope. The work described here provides the first upper bound for $\delta(TSP_n)$ that is independent of n. A tight lower bound for $\delta(TSP_n)$ remains an open question; i.e., does $\delta(TSP_n) = 2$, 3, or 4? Sierksma and Tijssen [7] determined by "brute force" that $\delta(TSP_n) = 2$, for $5 \le n \le 12$. For every $n > 12$, it is possible to construct tours such that exchanging perfect matchings requires three intermediate steps. Therefore, 4 is the best possible upper bound when simply exchanging perfect matchings.

As for perfect matching polytopes, our technique gives an immediate proof that $\delta(PM_n) = 2$, for every even $n \ge 8$. Using the construction given in section 3 of this paper, we can also prove the following constant upper bound for perfect 2-matching polytopes.

THEOREM 4. *$\delta(PTM_n) \le 6$, for every $n \ge 3$.*

Proof. It follows from Lemma 2 and the subtour patching technique (discussed often in [2]) that every extreme point of PTM_n either corresponds to a tour or is adjacent to an extreme point corresponding to a tour. We also know that from the construction given in section 3, every pair of extreme points on PTM_n corresponding to a pair of tours may be linked by a path on PTM_n of length at most 4. So given any pair of extreme points of PTM_n, we first move to extreme points corresponding to a pair of tours, if necessary, then join the tours with a path of length 4 or less. Hence, $\delta(PTM_n) \le 6$. □

REFERENCES

[1] V. CHVÁTAL, *On certain polytopes associated with graphs*, J. Combin. Theory Ser. B, 18 (1975), pp. 138–154.

[2] M. GRÖTSCHEL AND M. PADBERG, *Polyhedral theory*, in The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, E. Lawler, J.K. Lenstra, A. Rinnooy Kan, and D. Shmoys, eds., John Wiley, Chichester, 1986, pp. 251–305.

[3] V. KLEE AND P. KLEINSCHMIDT, *The d-step conjecture and its relatives*, Math Oper. Res., 12 (1987), pp. 718–755.

[4] M.W. PADBERG AND M.R. RAO, *The travelling salesman problem and a class of polyhedra of diameter two*, Math. Programming, 7 (1974), pp. 32–45.

[5] C.H. PAPADIMITRIOU, *The adjacency relation on the traveling salesman polytope is NP-complete*, Math. Programming, 14 (1978), pp. 312–324.

[6] F.J. RISPOLI, *The monotonic diameter of the perfect 2-matching polytope*, SIAM J. Optim., 4 (1994), pp. 455–460.

[7] G. SIERKSMA AND G. TIJSSEN, *Faces with large diameter on the symmetric traveling salesman polytope*, Oper. Res. Lett., 12 (1992), pp. 73–77.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 381–386, August 1998

003

ITERATING AN α-ARY GRAY CODE*

JONATHAN LICHTNER†

Abstract. In this paper we prove a theorem on the number of distinct codes produced when the α-ary Gray code mapping of Sharma and Khanna [*Inform. Sci.*, 15 (1978), pp. 31–43] is iteratively applied to an α-ary, dimension l code; that is, starting with an α-ary, dimension l code, and repeatedly applying the permutation given by Sharma and Khanna's mapping. From this theorem, it is easy to show there are $\Theta(l^q)$ distinct codes generated from this mapping, where q is the number of distinct primes in α (Let $f : \mathbf{N} \to \mathbf{R}^*$.$O(f)$ is the set of functions $g : \mathbf{N} \to \mathbf{R}^*$ such that for some $c \in \mathbf{R}^+$ and some $n_0 \in \mathbf{N}, g(n) \leq cf(n)$ for all $n \geq n_0$. $\Theta(f)$ is the set of functions $g : \mathbf{N} \to \mathbf{R}^*$ such that g is in $O(f)$ and f is in $O(g)$.). To prove this theorem we show that any base α, dimension l code word will cycle in $O(l^q)$ iterations of this Gray code mapping, and that this upper bound is attained. This theorem is a generalization of a theorem proven by Culberson [*Evolutionary Comput.*, 2 (1995), pp. 279–311] for the binary case.

Key words. Gray codes

AMS subject classification. 05E20

PII. S0895480196308736

1. Gray codes. The *Hamming distance* between two strings x and y of dimension, or length, l is the number of characters in which they differ. For example, the strings 34761 and 14781 have a Hamming distance of two. An α-ary Gray code of dimension l is a sequence of α^l distinct α-ary, length l strings such that any two adjacent code words have a Hamming distance of one. The Gray code discussed in this section will be represented by $\mathcal{G}(\alpha, l)$. We will sometimes use subscripts to refer to specific words in $\mathcal{G}(\alpha, l)$; that is, $\mathcal{G}_{i-1}(\alpha, l)$ is the ith word in $\mathcal{G}(\alpha, l)$. For example, $\mathcal{G}_0(2, 3) = 000$, $\mathcal{G}_1(2, 3) = 001$, and $\mathcal{G}_2(2, 3) = 011$.

A cyclic Gray code has the additional property that the Hamming distance between the first and last members in the Gray code is also one. $\mathcal{G}(\alpha, l)$ is a cyclic Gray code and has the special property that if $g = g_1 g_2 \dots g_l$ and $h = h_1 h_2 \dots h_l$ are cyclically adjacent numbers in $\mathcal{G}(\alpha, l)$ and $g_k \neq h_k$ (i.e., the characters that differ) then $h_k = g_k \oplus 1$. (We use the symbols $\oplus$ and $\ominus$ for addition mod α and subtraction mod α, respectively. As well, all summations in this paper are taken mod α.)

We can define $\mathcal{G}(\alpha, l)$ in terms of a function that maps the base α, dimension l integers, $\mathcal{N}(\alpha, l)$, to $\mathcal{G}(\alpha, l)$. For example, if x is the base α, dimension l representation of the nonnegative integer i, then this function will map x to $\mathcal{G}_i(\alpha, l)$. This mapping was given in [4], and we give the same mapping in a slightly altered form.

The mapping is denoted by $\mathcal{K}$, and its inverse by $\mathcal{K}^{-1}$. Let an element in $\mathcal{G}(\alpha, l)$ be represented by the string $g = g_1 g_2 \dots\ g_l$ and its corresponding base α integer be represented by the string $x = x_1 x_2 \dots x_l$. The mappings are then defined as

$$\mathcal{K}(x) = (g),$$
$$g_i = \begin{cases} x_1 & \text{if } i = 1, \\ x_i \ominus x_{i-1}, & 1 < i \leq l, \end{cases}$$

*Received by the editors September 3, 1996; accepted for publication (in revised form) June 26, 1997; published electronically July 7, 1998. This research was supported by Natural Sciences and Engineering Research Council (NSERC) grant OGP8053 and a Summer NSERC Scholarship.
http://www.siam.org/journals/sidma/11-3/30873.html
†66E. Brockington Crescent, Nepean, ON K2G 5L1, Canada (julich@nortel.ca).

EXAMPLE 1

$\mathcal{N}(2,3)$	$\mathcal{K}$	$\mathcal{G}(2,3)$
000	$\longmapsto$	000
001	$\longmapsto$	001
010	$\longmapsto$	011
011	$\longmapsto$	010
100	$\longmapsto$	110
101	$\longmapsto$	111
110	$\longmapsto$	101
111	$\longmapsto$	100

EXAMPLE 2

$\mathcal{N}(3,3)$	$\mathcal{K}$	$\mathcal{G}(3,3)$	$\mathcal{N}(3,3)$	$\mathcal{K}$	$\mathcal{G}(3,3)$
000	$\longmapsto$	000	112	$\longmapsto$	101
001	$\longmapsto$	001	120	$\longmapsto$	111
002	$\longmapsto$	002	121	$\longmapsto$	112
010	$\longmapsto$	012	122	$\longmapsto$	110
011	$\longmapsto$	010	200	$\longmapsto$	210
012	$\longmapsto$	011	201	$\longmapsto$	211
020	$\longmapsto$	021	202	$\longmapsto$	212
021	$\longmapsto$	022	210	$\longmapsto$	222
022	$\longmapsto$	020	211	$\longmapsto$	220
100	$\longmapsto$	120	212	$\longmapsto$	221
101	$\longmapsto$	121	220	$\longmapsto$	201
102	$\longmapsto$	122	221	$\longmapsto$	202
110	$\longmapsto$	102	222	$\longmapsto$	200
111	$\longmapsto$	100			

FIG. 1.1. *Two examples of Gray codes.*

and

$$\mathcal{K}^{-1}(g) = (x),$$
$$x_i = \begin{cases} g_1 & \text{if } i = 1, \\ g_i \oplus x_{i-1}, & 1 < i \le l, \end{cases}$$

$\mathcal{G}(2,l)$ is the binary reflected Gray code. Both $\mathcal{K}$ and $\mathcal{K}^{-1}$ can be computed in parallel. That is, g_i can be written in terms of x, and x_i can be written in terms of g. For instance, $x_i = g_1 \oplus g_2 \oplus \cdots \oplus g_i$. See Figure 1.1 for two examples of this Gray code. Further work on this Gray code was done in [5], and those interested in Gray codes in general may also wish to see [1] and [2].

2. Iterating the Gray code $\mathcal{G}(\alpha, l)$. In this section we prove a theorem on the number of distinct codes produced when $\mathcal{N}(\alpha,l)$ is iteratively mapped using $\mathcal{K}^{-1}$. That is, we start with $\mathcal{N}(\alpha,l)$ and repeatedly apply the permutation given by $\mathcal{K}^{-1}$. Let $\mathcal{N}_j^i(\alpha,l) = \mathcal{K}^{-1}(\mathcal{N}_j^{i-1}(\alpha,l))$ and $\mathcal{N}_j^0(\alpha,l) = \mathcal{N}_j(\alpha,l)$. Iteratively applying $\mathcal{K}^{-1}$ to each code word can be seen as iterating $\mathcal{G}(\alpha,l)$, since $\mathcal{N}^1(\alpha,l) = \mathcal{G}(\alpha,l)$.

We want to know the number of distinct codes that can be generated by iterating $\mathcal{N}(\alpha,l)$, or, more formally, for what $i > 0$ does $\mathcal{N}^i(\alpha,l) = \mathcal{N}(\alpha,l)$ such that $\forall j, 0 < j < i$, $\mathcal{N}^j(\alpha,l) \neq \mathcal{N}(\alpha,l)$?

The following theorem follows easily from Theorem 3.1 (proven in section 3).

THEOREM 2.1. *Let $l > 1$ and $\alpha = p_1^{n_1} p_2^{n_2} \dots p_q^{n_q}$, where p_i is prime, $p_i \neq p_j$ for $i \neq j$ (prime decomposition), and for each p_i, set h_i such that $p_i^{h_i - 1} < l \le p_i^{h_i}$. Then $\mathcal{K}^{-1}$ will generate $m = p_1^{h_1+n_1-1} p_2^{h_2+n_2-1} \dots p_q^{h_q+n_q-1}$ distinct codes.*

Proof. We know, from Theorem 3.1, that for any string x, $x^m = x$. We also know that this upper bound is attained for any string such that $x_1 \neq 0$, greatest common denominator (GCD) $(x_1, \alpha) = 1$, and $l > 1$. Since this is true for some strings (e.g., any string whose first character is 1), we know that iterating $\mathcal{N}(\alpha,l)$ gives m distinct codes. ∎

If $l = 1$ then $m = 1$; for $l > 1$, $l^q \le m < \alpha l^q$, where q is the number of distinct primes in α. This implies that the maximum number of codes generated is $\Theta(l^q) \forall l$.

3. Iterating strings using $\mathcal{K}^{-1}$. In this section we will prove a theorem on the cycles induced when an α-ary, dimension l string is iterated. We will use the notation

$x^i = \mathcal{K}^{-i}(x) = \mathcal{K}^{-1}(\mathcal{K}^{-(i-1)}(x))$ and $x^0 = \mathcal{K}^0(x) = x$. We use subscripts to refer to a particular digit in x. For example, if $x = x^0 = 12356$, then $x_3^0 = 3$.

A cycle on string x consists of the sequence $x^0, x^1, \ldots, x^{i-1}$, where $x^i = x^0$ and $\forall j$ such that $0 < j < i, x^j \neq x^0$. The following cycle theorem gives an upper bound on the cycle length of any string x, and gives the actual cycle length when $\mathrm{GCD}(x_1, \alpha) = 1$ and $x_1 \neq 0$.

THEOREM 3.1. *Let $l > 1$ and $\alpha = p_1^{n_1} p_2^{n_2} \ldots p_q^{n_q}$, where p_i is prime, $p_i \neq p_j$ for $i \neq j$ (prime decomposition), and for each p_i, set h_i such that $p_i^{h_i - 1} < l \leq p_i^{h_i}$. If $m = p_1^{h_1+n_1-1} p_2^{h_2+n_2-1} \ldots p_q^{h_q+n_q-1}$, then $x^m = x$, and if $x_1 \neq 0$ and GCD$(x_1, \alpha) = 1$, then for any $0 < m' < m$, $x^{m'} \neq x$.*

Given any α-ary string and the number m as described in Theorem 3.1, mapping the string with the inverse Gray code mapping m times will cause the iterated string to return to its original value. If $x_1 \neq 0$ and $\mathrm{GCD}(x_1, \alpha) = 1$, then m is the smallest integer for which the code word will cycle. For example, if $x = 1000$ and $\alpha = 2$, then $x^1 = 1111$, $x^2 = 1010$, $x^3 = 1100$, $x^4 = 1000$, which implies that $m = 4$. The special case of this theorem for $\alpha = 2$ was proven in [3].

Before proving this theorem, we will prove a number of lemmas.

LEMMA 3.2.

$$x_j^i = x_{j-1}^i \oplus x_j^{i-1}, \qquad i > 0, \quad 1 < j \leq l.$$

Proof. The proof is from the definition of $\mathcal{K}^{-1}$. □

LEMMA 3.3.

$$x_j^1 = x_1 \oplus x_2 \oplus \cdots \oplus x_j, \qquad 1 < j \leq l,$$

and

$$x_1^i = x_1, \qquad i \geq 1.$$

Proof. Both statements follow from Lemma 3.2. □

LEMMA 3.4.

$$x_j^i = \sum_{k=1}^{j} \binom{i+j-k-1}{j-k} x_k, \qquad 1 \leq j \leq l, \quad i \geq 1.$$

Proof.

Basis: Lemma 3.3 is the basis, and it can be seen that both of the statements in Lemma 3.3 are special cases of Lemma 3.4.

Induction step:

I.H: Assume Lemma 3.4 is true for x_{j-1}^i and x_j^{i-1}.

We know from Lemma 3.2 that

$$x_j^i = x_{j-1}^i \oplus x_j^{i-1},$$

and applying the induction hypothesis yields

$$x_j^i = \sum_{k=1}^{j-1} \binom{i+j-k-2}{j-k-1} x_k \oplus \sum_{k=1}^{j} \binom{i+j-k-2}{j-k} x_k.$$

We now add the kth terms, $1 \leq k \leq j-1$, to obtain

$$\binom{i+j-k-2}{j-k-1} x_k \oplus \binom{i+j-k-2}{j-k} x_k,$$

which is equal to

$$\binom{i+j-k-1}{j-k} x_k,$$

which satisfies Lemma 3.4. The x_j term occurs only once, and it also satisfies Lemma 3.4. □

We now introduce the notation $c^i_{j,k}$, where $1 \leq k \leq j$. $c^i_{j,k}$ refers to the coefficient of the kth term of the equation given in Lemma 3.4 for x^i_j. That is

$$c^i_{j,k} = \binom{i+j-k-1}{j-k}.$$

LEMMA 3.5. *Let $x = x_1 x_2 \dots x_l$ be any α-ary string. If $\forall j, 2 \leq j \leq l, c^m_{j,1} = 0 (\text{mod } \alpha)$, then $x^m = x$.*

Proof. To show $x^m = x$ we must show that $\forall j, 1 \leq j \leq l, x^m_j = x_j$. Note that $x^m_1 = x_1$; thus we need only consider an arbitrary j, $2 \leq j \leq l$.

Assume $\forall j', 2 \leq j' \leq l, c^m_{j',1} = 0 (\text{mod } \alpha)$. Then

$$\begin{aligned} x^m_j &= \sum_{k=1}^{j} c^m_{j,k} x_k \\ &= \sum_{k=1}^{j-1} c^m_{j,k} x_k \oplus c^m_{j,j} x_j \\ &= \sum_{k=1}^{j-1} c^m_{j,k} x_k \oplus x_j, \end{aligned}$$

since $c^m_{j,j} = 1$. But since

$$c^m_{j.k} = c^m_{j-k+1,1},$$

the summation is equal to $0 (\text{mod } \alpha)$, and $x^m_j = x_j$. □

LEMMA 3.6. *Let $x = x_1 x_2 \dots x_l$ be any α-ary string such that $x_1 \neq 0$ and GCD$(x_1, \alpha) = 1$. If $\exists j, 2 \leq j \leq l$ such that $c^m_{j,1} \neq 0 (\text{mod } \alpha)$, then $x^m \neq x$.*

Proof. Assume $x_1 \neq 0$, $\text{GCD}(x_1, \alpha) = 1$, and $c^m_{j,1} \neq 0 (\text{mod } \alpha)$, for some $j, 2 \leq j \leq l$. If $c^m_{2,1} \neq 0 (\text{mod } \alpha)$, then $x^m_2 = m x_1 \oplus x_2 \neq x_2$ and we are done. Otherwise, $c^m_{j,1} \neq 0 (\text{mod } \alpha)$, for some j, $2 < j \leq l$ and that $c^m_{j',1} = 0 (\text{mod } \alpha)$, for $j', 2 \leq j' < j$. Then

$$x^m_j = c^m_{j,1} x_1 \oplus c^m_{j,2} x_2 \oplus \dots \oplus c^m_{j,k-1} x_{j-1} \oplus x_j,$$

but note that

$$c^m_{j,k} = c^m_{j-k+1,1},$$

which means that

$$x^m_j = c^m_{j,1} x_1 \oplus x_j,$$

since all the other terms are congruent to zero. Since $c_{j,1}^{m} \neq 0(\text{mod } \alpha)$, then $x_j^m \neq x_j$. □

Lemma 3.5 shows that finding an m that sets $c_{j,1}^{m} = 0(\text{mod } \alpha)$ for $2 \leq j \leq l$ implies that $x^m = x$. Lemma 3.6 shows that if we choose the smallest such m that sets $c_{j,1}^{m} = 0(\text{mod } \alpha)$ for $2 \leq j \leq l$ and $x_1 \neq 0$ and $\text{GCD}(x_1, \alpha) = 1$, then $x^{m'} \neq x$, for $0 < m' < m$. Our goal then will be to set the $c_{j,1}^{m}$ terms to zero (mod α), picking the smallest such m that would do so.

We now prove Theorem 3.1. To do this we use induction on j, and within the inductive proof, we will use the fact that $c_{j,1}^{m} = \frac{m+j-2}{j-1} c_{j-1,1}^{m}$. Since m increases with l, we will use the notation m_j, which refers to the cycle upper bound length on strings of length j. If $j > i$, then m_j will be a multiple of m_i. Let p_i be a prime in α. Since h_i also varies with l, we use the notation $h_{i,j}$ within the proof.

Proof of Theorem 3.1. Basis ($j = 2$):

$$c_{2,1}^{m_2} = m_2 \quad \forall m_2 > 1.$$

We pick the minimum m_2 that will set $m_2 = 0(\text{mod } \alpha)$, or $m_2 = \alpha = p_1^{n_1} p_2^{n_2} \dots p_q^{n_q}$.

It is easy to see that Theorem 3.1 is satisfied for $j = 2$ and that p_i occurs n_i times in the prime-power factorization of $c_{2,1}^{m_2}$, which corresponds to I.H.2 (defined below). In this case $h_{i,2} = 0$.

Induction step ($j > 2$):

For the induction step we need only consider an arbitrary prime $p_i, 1 \leq i \leq q$.

I.H.1: Assume setting m_{j-1} as in Theorem 3.1 will set $c_{j',1}^{m_{j-1}} = 0(\text{ mod } \alpha), \forall j', 2 \leq j' < j$.

I.H.2: Assume $j - 1 = C_1 p_i^d + 1$ (where p_i and C_1 are relatively prime, $d < h_{i,j-1}$) implies p_i occurs $n_i + h_{i,j-1} - 1 - d$ times in the prime-power factorization of $c_{j-1,1}^{m_{j-1}}$.

I.H.2 is needed because we must know how many factors of p_i are in $c_{j-1,1}^{m_{j-1}}$. If we know this, then using the fact that $c_{j,1}^{m_{j-1}} = \frac{m_{j-1}+j-2}{j-1} c_{j-1,1}^{m_{j-1}}$, we can determine the number of times p_i occurs in the prime-power factorization of $c_{j,1}^{m_{j-1}}$. If there are n_i or more such occurrences, then $c_{j,1}^{m_{j-1}} = 0(\text{mod } \alpha)$ and it is sufficient to set $m_j = m_{j-1}$; otherwise, m_j must be increased (while still being a multiple of m_{j-1}). This leads to two cases:

1. $j = C_2 p_i^{d'} + 1, 0 \leq d' < h_{i,j-1}$,
2. $j = p_i^{d'} + 1, d' = h_{i,j-1}$,

where C_2 and p_i are relatively prime. For each case we must now show that I.H.1 and I.H.2 hold for j.

Case 1 ($j = C_2 p_i^{d'} + 1, 0 \leq d' < h_{i,j-1}$). Recall that $c_{j,1}^{m_{j-1}} = \frac{m_{j-1}+j-2}{j-1} c_{j-1,1}^{m_{j-1}}$. In this case, p_i will occur d times in the prime-power factorization of $m_{j-1} + j - 2$ and p_i will occur d' times in the prime-power factorization of $j - 1$, and the total number of factors of p_i will be $n_i + h_{i,j-1} - 1 - d + d - d' = n_i + h_{i,j-1} - 1 - d'$. Setting $m_j = m_{j-1}$ (and $h_{i,j} = h_{i.j-1}$) corresponds to Theorem 3.1; I.H.1 holds for j since we need at least n_i occurrences of p_i in $c_{j1}^{m_{j-1}}$, and this is the case. I.H.2 also holds.

Case 2 ($j = p_i^{d'} + 1, d' = h_{i,j-1}$). In this case it can be easily seen that p_i occurs $n_i - 1$ times in the prime-power factorization of $c_{j,1}^{m_{j-1}}$, but that n_i are needed. Setting $m_j = p_i m_{j-1}$ (as in Theorem 3.1, i.e., $h_{i,j} = h_{i,j-1} + 1$) will make $c_{j,1}^{m_j}$ have one more factor of p_i than $c_{j,1}^{m_{j-1}}$, while leaving all other factors unchanged. Then,

$$c_{j,1}^{m_j} = \frac{(m_j + j - 2)(m_j + j - 3) \cdots (m_j)}{(j-1)!}$$

and

$$c_{j,1}^{m_{j-1}} = \frac{(m_{j-1}+j-2)(m_{j-1}+j-3)\cdots(m_{j-1})}{(j-1)!}.$$

The denominator of $c_{j,1}^{m_j}$ is equal to that of $c_{j,1}^{m_{j-1}}$, and so the denominators have the same number of factors of p_i. Thus we need consider only the numerator. Consider an arbitrary factor of the numerator of $c_{j,1}^{m_j}$, $m_j + k$, where $0 \le k \le j-2$. In the $k = 0$ case, $c_{j,1}^{m_j}$ has an extra factor of p_i. When k is nonzero, there are no extra factors of $p_s, 1 \le s \le q$. There are two cases to consider, $p_s \ne p_i$ and $p_s = p_i$. For the first case, $j - 2 < p_s^{h_{s,j}}$ which means that $m_j + k = (p_s E + F)p_s^t$ and $m_{j-1} + k = (p_s E' + F')p_s^t$ where F and F' have no factors of p_s, and E, E', and t are constants, and thus both I.H.1 and I.H.2 are satisfied. For the latter case a similar argument suffices but uses the fact that $j - 2 < p_i^{h_{i,j-1}}$ (since $j = p_i^{h_{i,j-1}} + 1$). □

When $x_1 \ne 0$ or $\mathrm{GCD}(x_1, \alpha) \ne 1$, the m of Theorem 3.1 may be larger than the cycle length of x (though m will be a multiple of x's cycle length). For an example x where Theorem 3.1 describes the cycle length, consider $x = 123456$ for $\alpha = 10$. In this case $m = 200$, as the theorem states. The strings 421 and 4211 for $\alpha = 8$ are two examples of strings whose minimum cycle lengths are 2, which is less than the $m = 16$ of Theorem 3.1.

4. Conclusion. In this paper we discussed the problem of iteratively applying the inverse Gray code mapping to strings, and showed that a cycle on any string x will have length in $O(l^q)$, where l is the length of x and q is the number of distinct primes in α. If $\mathrm{GCD}(x_1, \alpha) = 1$ and $x_1 \ne 0$, then x will have a cycle length in $\Theta(l^q)$. This implies that the number of distinct codes generated by iterating $\mathcal{N}(\alpha, l)$ (or any α-ary, dimension l code) using $\mathcal{K}^{-1}$ is $\Theta(l^q)$.

Acknowledgments. The author thanks Joseph Culberson for his comments on earlier versions of this paper, and the author would also like to thank the anonymous referees.

REFERENCES

[1] L. S. Barasch, S. Lakshmivarahan, and S. K. Dhall, *Generalized Gray codes and their properties*, in Mathematics for Large Scale Computing, Lecture Notes in Pure and Appl. Math. 120, J.C. Diaz, ed., Marcel Dekker, New York, 1989, pp. 203–216.

[2] J. R. Bitner, G. Ehrlich, and E. M. Reingold, *Efficient generation of the binary reflected Gray code and its applications*, Comm. ACM, 19 (1976), pp. 517–521.

[3] J. Culberson, *Mutation-crossover isomorphisms and the construction of discriminating functions*, Evolutionary Comput., 2 (1995), pp. 279–311.

[4] B. D. Sharma and R. K. Khanna, *On m-ary Gray codes*, Inform. Sci., 15 (1978), pp. 31–43.

[5] B. D. Sharma and R. K. Khanna, *Integer characterization of binary and m-ary Gray codes*, J. Combin. Inform. System Sci., 4 (1979), pp. 227–236.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 387–399, August 1998

004

APPROXIMATE CORE ALLOCATION FOR BINPACKING GAMES*

ULRICH FAIGLE† AND WALTER KERN†

Abstract. A *binpacking game* is a cooperative N-person game, where the set of players consists of k bins of size 1 and n items of sizes $a_1, \ldots, a_n$. The value of a coalition of bins and items is the maximum total size of items in the coalition that can be packed into the bins of the coalition. Our main result asserts that for every $\epsilon > 0$, there exist ϵ-approximate core allocations provided k is large enough. Moreover, for every fixed $\delta > 0$, the smallest ϵ for which the ϵ-approximate core of a given binpacking game is nonempty can be computed in polynomial time with error at most δ, provided k is sufficiently large. We furthermore derive more specialized results for some subclasses of binpacking games.

Key words. binpacking, core, cost, N-person game

AMS subject classifications. 90C27, 90D12

PII. S0895480296308074

1. Introduction.

A *cooperative (maximum value) N-person game* is defined by a set N of *players* and a *characteristic* (or *value*) *function* $v : 2^N \to \mathbb{R}$ satisfying $v(\emptyset) = 0$. A subset $S \subseteq N$ is called a *coalition* and N itself is the *grand* coalition. In the usual interpretation, $v(S)$ is taken to represent the gain that coalition S can achieve if all its members cooperate.

A central question of cooperative game theory is how to allocate the total gain $v(N)$ among the individual players $i \in N$ in a "fair" way. One of the most attractive allocation concepts goes back to von Neumann and Morgenstern [10] attempting to allocate a vector in the core of the game (see also Shapley [11, 12]).

Here we define the *core* core(v) of our game to be the polytope of all vectors $x \in \mathbb{R}^N$ satisfying

(i) $x(N) \le v(N)$,

(ii) $x(S) \ge v(S)$ for all $S \subseteq N$,

where we use the notation $x(S) = \sum_{i \in S} x_i$.

Because a game may have an empty core, one might be tempted to relax condition (ii) above in such a way that the modified core becomes nonempty. Several ways of doing so have been proposed in the literature (see, for example, the discussion and the references in Faigle and Kern [3]). One of these concepts was introduced in Faigle and Kern [3] as the (multiplicative) ϵ-core and further expanded to the concept of the *nucleon* in Faigle et al. [5]. (For a survey of other and/or related allocation concepts see, e.g., Weber [15]).

Given $\epsilon \ge 0$, the *ϵ-core* ϵ-core(v) is defined as the polytope of all vectors $x \in \mathbb{R}^N$ satisfying condition (i) above together with condition

(ii′) $x(S) \ge (1 - \epsilon)v(S)$ for all subsets $S \subseteq N$.

If the characteristic function v is nonnegative (which will be the case for all binpacking games we consider here), 1-core(v) is obviously nonempty. In order to have an allocation concept that approximates the core as closely as possible, one now would like to have ϵ as small as possible with the guarantee that ϵ-core(v) is nonempty.

*Received by the editors February 22, 1996; accepted for publication (in revised form) September 15, 1997; published electronically July 7, 1998.
http://www.siam.org/journals/sidma/11-3/30807.html

†Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands (faigle@math.utwente.nl, kern@math.utwente.nl).

There are many interesting classes of games whose characteristic functions are implicitly given as optimum values of combinatorial optimization problems (cf. Tijs and Borm [13]). In this context, the question naturally arises of whether it is possible to efficiently compute good ϵ-core allocations for particular classes of combinatorial games. Our aim here is to study classes of binpacking games as introduced in Faigle and Kern [3].

A *binpacking game* is defined by a set of k bins, each of size (or *capacity*) 1, and n items $1, 2, \ldots, n$ of sizes (or *weights*) $a_1, \ldots, a_n$, where we assume, w.l.o.g, $0 \leq a_i \leq 1$. The player set N consists of all bins and all items. So we have $|N| = n + k$.

The value $v(S)$ of a coalition S containing $k' \leq k$ bins and items $a_{i_1}, \ldots, a_{i_s}$ is the weight of an optimal binpacking relative to S, i.e.,

$$v(S) := \max \sum_{j=1}^{k'} \sum_{i \in I_j} a_i,$$

where the maximum is taken over all collections of pairwise disjoint subsets $I_1, \ldots, I_{k'} \subseteq \{i_1, \ldots, i_s\}$ such that

$$\sum_{i \in I_j} a_i \ \leq \ 1.$$

We set $v(S) = 0$ if $k' = 0$ or $s = 0$.

Example. Consider the binpacking game with $k = 2$ bins, three items of size $a_1 = a_2 = a_3 = 1/2$, and two items of size $a_4 = a_5 = 1/2 + \delta$ for some $\delta > 0$. Letting $\delta \to 0$, it is shown in Faigle and Kern [3] that for each $\epsilon < 1/7$, there is a binpacking game with empty ϵ-core.

It was shown in Faigle and Kern [3] that the 1/2-core of a binpacking game is always nonempty. Woeginger [16] improved this result to $\epsilon = 1/3$. Kuipers [7] showed that the 1/7-core of a binpacking game is nonempty if all weights a_i are *strictly* larger than 1/3 (see also section 2 below).

An intriguing conjecture has been proposed by Woeginger [17]:

Conjecture 1. There exists a universal constant $C > 0$ such that each binpacking game v admits an allocation vector $x \in \mathbb{R}^N$ with the properties

(i′) $x(N) \ \leq \ v(N) + C$;

(ii) $x(S) \ \geq \ v(S)$ for all subsets $S \subseteq N$.

A weaker conjecture is the following.

Conjecture 2. For every $\epsilon > 0$, there exists a constant $K(\epsilon)$ such that each binpacking game with $k \geq K(\epsilon)$ bins admits a nonempty ϵ-core.

It is not hard to see that Conjecture 1 implies Conjecture 2. Indeed, for any given $\epsilon > 0$, take $K(\epsilon) = 2C/\epsilon$ and consider a binpacking game with $k \geq 2C/\epsilon$ bins. If all items $a_1, \ldots, a_n$ fit into the k bins, then there exists a trivial core allocation: allocate a_i to the ith item and *zero* to every bin.

If it is impossible to pack all items, an optimal packing fills each bin to at least half its capacity. Thus $v(N) \geq k/2$. Choose now the allocation vector $x \in \mathbb{R}^N$ as in Conjecture 1 and observe $x(N) \leq (1 + \epsilon)v(N)$.

Let $x^\epsilon = (1 + \epsilon)^{-1}x$. Then $x^\epsilon(N) \leq v(N)$ and for every $S \subseteq N$,

$$x^\epsilon(S) \geq (1 + \epsilon)^{-1}v(S) \geq (1 - \epsilon)v(S),$$

i.e., x^ϵ is a member of the ϵ-core and Conjecture 2 holds.

The main purpose of the present paper is to prove Conjecture 2. The rest of the paper is organized as follows. In section 2, we introduce some notation and relate the allocation problem to the problem of the duality gap with respect to a certain linear programming relaxation of the binpacking problem. Section 3 proves Conjecture 1 for two particular subclasses of binpacking games (one of them being the class of games with all items strictly larger than 1/3). Section 4 establishes the proof of Conjecture 2. A main tool here is the combination of the results of section 2 with the binpacking technique introduced by de la Vega and Lueker [14]. In section 5, we investigate the problem of computing the smallest ϵ for which a given binpacking game has a nonempty ϵ-core. We show that, for every fixed $\delta > 0$, the optimal ϵ can be determined within error δ in polynomial time, provided k is strictly larger than $1/\delta$.

We would like to point out that the binpacking problem we are facing here is closely related to the "usual" binpacking problem but differs in an important aspect: one usually wants to *minimize the number of bins* so that the given items can be packed, while our problem asks us to *maximize the (weighted) number of items* that can be packed into the given number of bins.

2. The duality gap. We approach Conjecture 1 by studying the linear program

$$\begin{array}{lllll} (LP) & \min & x(N), & & \\ & \text{s.t.} & x(S) & \geq & v(S) \qquad \text{for all } S \subseteq N. \end{array}$$

(Note that $x \geq 0$ is implied by the constraints arising from the singleton coalitions.)

We want to bound the difference $x^*(N) - v(N)$, where x^* is an optimal solution of (LP) above. Note that $x^*(N) - v(N) \leq 0$ holds if and only if $\text{core}(v) \neq \emptyset$. In general, this difference can be interpreted as a duality gap as we will see in the following. (A more general framework for the study of this duality gap is the model of "partition games" as discussed in Faigle and Kern [4]).

To simplify the notation, note that, by symmetry, there exists an optimal solution x^* allocating the same amount x_0^* to each bin. Furthermore, it apparently suffices to consider only those restrictions $x(S) \geq v(S)$ where S consists of (exactly) one bin and some subset $I \subseteq \{1, \ldots, n\}$ of items with total size

$$\sum_{i \in I} a_i \leq 1.$$

Let us call such an $I \subseteq \{1, \ldots, n\}$ *feasible* and denote by $\mathcal{F}$ the collection of all feasible subsets of items.

Denote by $\sigma = (\sigma_I) \in \mathbb{R}^{\mathcal{F}}$ the *total size vector*, i.e.,

$$\sigma_I = \sum_{i \in I} a_i \quad \text{for all } I \in \mathcal{F}.$$

Now our allocation problem can be written in the form

$$\begin{array}{lllllll} (AP) & \min & kx_0 & + & \sum_{i=1}^{n} x_i, & & \\ & \text{s.t.} & x_0 & + & x(I) & \geq & \sigma_I \qquad \text{for all } I \in \mathcal{F}, \\ & & & & x_0, x & \geq & 0. \end{array}$$

Its dual is a "fractional packing problem":

$$\begin{array}{lll} (PP) & \max\ \sigma^T y, & \\ & \text{s.t.} \quad \sum_{I\in\mathcal{F}} y_I \le k, & \\ & \qquad \sum_{I\ni i} y_I \le 1 & (i=1,\dots,n), \\ & \qquad\quad y \ge 0. & \end{array}$$

The integer linear program corresponding to (PP) is

$$\begin{array}{lll} (IPP) & \max\ \sigma^T y, & \\ & \text{s.t.} \quad \sum_{I\in\mathcal{F}} y_I \le k, & \\ & \qquad \sum_{I\ni i} y_I \le 1 & (i=1,\dots,n), \\ & \qquad\quad y \in \{0,1\}^{\mathcal{F}}. & \end{array}$$

Observe that (IPP) is just an integer linear programming formulation of our binpacking problem given by k bins and items with weights $a_1,\dots,a_n$. Thus the optimal value of (IPP) is $v(N)$ and, by linear programming duality, the optimal value of (PP) equals the optimal value $x^*(N)$ of our original problem (LP).

Conjecture 1 thus states that the duality gap $x^*(N)-v(N)$ can be bounded by a universal constant $C>0$.

There is a direct relationship between the duality gap and the nonemptyness of the ϵ-core.

LEMMA 2.1. *ϵ-core$(v)\neq\emptyset$ if and only if $\epsilon \ge (x^*(N)-v(N))/x^*(N)$.*

Proof. Note that $x\in\epsilon$-core(v) holds if and only if $x(N)\le v(N)$ and the vector $(1-\epsilon)^{-1}x$ is a feasible solution for the linear program (LP). □

We remark that proving Conjecture 1 would be of no help from the point of view of practical computation, since there is no obvious way to solve (PP). Yet, the situation is computationally not quite hopeless; we show in section 5 that, for every fixed error bound $\delta>0$, the smallest ϵ with ϵ-core$(v)\neq\emptyset$ can be approximated within δ in polynomial time if $k>1/\delta$.

3. Two special cases. In this section, we will prove Conjecture 1 for two special classes of binpacking games. In the first class, the item sizes a_i are required to be *strictly* larger than $1/3$. So each feasible set of items has at most two members. Our binpacking problem thus reduces to a weighted matching problem, which we analyze in the spirit of the argument Kuipers [7] employed as a refinement of an argument in Faigle and Kern [3].

We will use a generalization of a well-known half-integrality result for perfect matchings. Let $G=(V,E)$ be a graph on the set V of nodes and an edge weighting $w: E\to\mathbb{R}$, and consider the linear program

$$\begin{array}{lll} (FM) & \max\ w^T y, & \\ & \text{s.t.} \quad \sum_{e\in E} y_e \le k, & \\ & \qquad \sum_{e\ni i} y_e \le 1 & (i=1,\dots,|V|), \\ & \qquad\quad y_e \ge 0. & \end{array}$$

LEMMA 3.1. *There exists an optimal solution* $y^* \in \{0, 1/2, 1\}^E$ *such that the set*

$$\mathcal{C} = \{e \in E \mid y_e^* = 1/2\}$$

is a disjoint union of circuits of odd size and at most one single edge.

Proof. It is well known that (FM) has an optimal solution $y^* \in \{0, 1/2, 1\}^E$ (cf. Lovász and Plummer [8]). Let C be any connected set of edges in $\mathcal{C}$. If C is an even circuit or a path with at least two edges, then either the even edges or the odd edges of C have total weight of at least $w(C)/2$. So we may modify y^* by either setting the even edges in C to value 1 (and the odd edges to 0) or conversely in order to obtain a feasible solution $\hat{y}$ for (FM) yielding the same objective function value.

If there are two single edges in $\mathcal{C}$, we modify y^* by setting the value of the larger one to 1 and of the smaller to 0, and the lemma follows. □

THEOREM 3.2. *If all item sizes* $a_1, \ldots, a_n$ *in the binpacking game are strictly larger than* $1/3$*, then the duality gap is at most* $1/4$.

Proof. Note that the duality gap of a binpacking game is 0 whenever $k \geq n$. We therefore assume $k < n$. Consider now a game given by k bins and items of sizes $a_1, \ldots, a_n$, where $a_i > 1/3$. We construct a weighted graph $G = (V, E)$ on $|V| = 2n$ nodes as follows:

(a) Each node $i = 1, \ldots, n$ is assigned weight $w_i = a_i$.
(b) Each node $i = n+1, \ldots, 2n$ is assigned weight $w_i = 0$.
(c) Let $E = \{(i, j) \mid w_i + w_j \leq 1\}$.
(d) For each $e \in E$, let $w_{ij} = w_i + w_j$ be the weight of e.

The optimization problems (IPP) and (PP) from section 2 can now be restated as

$$\begin{array}{lll} (IPP) & \max \ w^T y, & \\ & \text{s.t.} \quad \sum_{e \in E} y_e \leq k, & \\ & \qquad \sum_{e \ni i} y_e \leq 1 & (i = 1, \ldots, 2n), \\ & \qquad\quad y_e \in \{0, 1\}, & \end{array}$$

and

$$\begin{array}{lll} (PP) & \max \ w^T y, & \\ & \text{s.t.} \quad \sum_{e \in E} y_e \leq k, & \\ & \qquad \sum_{e \ni i} y_e \leq 1 & (i = 1, \ldots, 2n), \\ & \qquad\quad y_e \geq 0. & \end{array}$$

(IPP) is a (restricted cardinality) maximum weighted matching problem and (PP) is the corresponding fractional relaxation (FM) above. Lemma 3.1, therefore, guarantees us the existence of an optimal fractional solution $y^* \in \{0, 1/2, 1\}^E$ such that the set

$$\mathcal{C} = \{e \in E \mid y_e^* = 1/2\}$$

is a disjoint union of odd circuits and at most one single edge.

Claim. We may assume that $\mathcal{C}$ contains at most one odd circuit.

To see the validity of the claim, suppose there are two disjoint odd circuits C_1 and C_2. Choose nodes $i \in C_1$ and $j \in C_2$ of minimal weight respectively. Then $w_i \leq 1/2$ and $w_j \leq 1/2$, and hence $e = (i,j) \in E$. It is now clear how to modify y^* on $C_1 \cup C_2 \cup e$ to a fractional solution $\hat{y}$ with the same weight so that $\hat{y}$ takes on values in $\{0,1\}$ on $C_1 \cup C_2 \cup e$. Repeating this argument, we arrive at the desired optimal solution with at most one odd circuit.

Now choose y^* as to satisfy the claim and let C be the odd circuit in $\mathcal{C}$. If C contains a node v, say, of weight $w(v) = 0$, we can easily modify y^* to a feasible optimal solution $\hat{y}$ that takes on $(0,1)$-values on the edges of C. Because k is an integer, $\hat{y}$ can be assumed to have *no* edge of value $1/2$, which implies that $\hat{y}$ is also feasible for (IPP) and that the duality gap is 0.

In the remaining case, we assume y^* to be such that $\mathcal{C}$ contains exactly one odd circuit C with each node of positive weight. There is also no loss of generality, assuming $y^*_e = 0$ whenever $w_e = 0$. Because G contains at least n nodes of weight 0, $\sum_{e \in C} y^*_e > 1$ and $k < n$ then imply that there must exist some node $v \in V$ such that $w_v = 0$ and $y^*_e = 0$ for every edge e with endpoint v. Let (i,j) be an edge of C with $w_i = a_i \leq 1/2$. We modify y^* to be feasible solution $\bar{y}$ by setting $\bar{y}_{(v,j)} = 1/2$ and $\bar{y}_{(i,j)} = 0$. Observe

$$w^T y^* - w^T \bar{y} = a_i/2 \leq 1/4.$$

Since $\{e \in E \mid \bar{y} = 1/2\}$ contains no odd circuit, we can modify $\bar{y}$ as before to a feasible solution $\hat{y}$ for (IPP) with $w^T \bar{y} \geq w^T \hat{y}$, which proves the theorem. □

Our second result bounds the duality gap by a constant $C = C(m)$ for the class of binpacking games where the number of distinct item *sizes* is not more than m (the number n of items, of course, may be larger than m). This result will be of further use in section 4.

THEOREM 3.3. *If the item sizes $a_1, \ldots, a_n$ take on at most m different values, then the duality gap is bounded by $C(m) = m/2$.*

Proof. Assume that the item sizes are $a_1, \ldots, a_m$ and occur with multiplicities $\mu_1, \ldots, \mu_m$. Then each feasible set $I \in \mathcal{F}$ can be described by its *type vector* $T = (t_1, \ldots, t_m)$ indicating the number t_i of items of size a_i that occur in I. Let $\mathcal{T}$ be the set of feasible types and let

$$\sigma_T = \sum_{i=1}^{m} t_i a_i.$$

Then the problems (IPP) and (PP) from section 2 are equivalent to

$$\begin{aligned}
(IPP') \quad \max \ & \sigma^T z, \\
\text{s.t.} \quad & \sum_{T \in \mathcal{T}} z_T \leq k, \\
& \sum_{T=(t_1,\ldots,t_m) \in \mathcal{T}} t_i z_T \leq \mu_i \qquad (i = 1, \ldots, m), \\
& z_T \in \mathbb{N}_0,
\end{aligned}$$

and

$$
\begin{aligned}
(PP') \quad \max \ & \sigma^T z, \\
\text{s.t.} \quad & \sum_{T \in \mathcal{T}} z_T \leq k, \\
& \sum_{T=(t_1,\ldots,t_m)\in\mathcal{T}} t_i z_T \leq \mu_i \qquad (i = 1, \ldots, m), \\
& z_T \geq 0.
\end{aligned}
$$

(Indeed, the equivalence of (IPP') with (IPP) is straightforward to verify, for example, by induction on the μ_i's. The equivalence of (PP') with (PP) follows from the observation that their linear programming duals are equivalent.)

Assuming, w.l.o.g., $k \geq 1$ and $\mu_i \geq 1$ for all $i = 1, \ldots, m$, the feasibility region of (PP') is full-dimensional (because $z_L = \epsilon$ yields a feasible solution for every sufficiently small $\epsilon > 0$). Hence every optimal basic solution z^* satisfies at least $|\mathcal{T}|$ restrictions with equality, which in turn yields $|\text{supp}(z^*)| \leq m + 1$. Rounding down each component of z^* to the nearest integer results in a feasible solution $\hat{z}$ for (IPP). Let

$$\Delta = k - \sum_{T \in \mathcal{T}} \hat{z}_T,$$

and note that $z' = z^* - \hat{z}$ must be an optimal fractional packing of the remaining items into Δ bins.

Hence, if $\hat{z} \neq 0$, we may argue by induction on the number n of items that there exists an integral packing z'' of the remaining items into Δ bins with a duality gap of at most $C(m)$, which establishes the bound $C(m)$ on the duality gap of the original problem.

If $\hat{z} = 0$ we construct an integral packing z'' "greedily" by filling the first bin with items to the largest weight possible, then the second, etc.. This packing guarantees that the first bin is filled with items of total weight at least

$$\sigma_{max} := \max\{\sigma_T \mid z^*_T \neq 0\}.$$

If z'' uses all of the available items, then $\sigma^T z'' \geq \sigma^T z^*$, and the duality gap is 0. Otherwise, z'' fills each bin to weight at least $1/2$. So

$$\sigma^T z'' \geq \sigma_{max} + (k-1)/2.$$

Now $\text{supp}(z^*) \leq m+1$ yields $\sigma^T z^* \leq \sigma_{max} + m$. Hence $m < k$ implies $\sigma^T z^* - \sigma^T z'' \leq m/2$.

If $m \geq k$, $\sigma^T z^* \leq \sum_{T\in\mathcal{T}} z^* \leq k$ and $\sigma^T z'' \geq k/2$ imply the claimed bound on the duality gap. □

4. Arbitrary item sizes. We approach the binpacking game with arbitrary item sizes with the binpacking method of de la Vega and Lueker [14]. We slightly modify the item sizes so that they take on a relatively small number m of distinct values only. We then estimate the change in the objective function value resulting from this modification. Taking Theorem 3.3 into account, we finally are able to bound the duality gap.

To fix notation, assume we are given a binpacking game with k bins and items $a_1, \ldots, a_n$. We always consider the items as presented in an ordered list

$$L \ : \ a_1 \leq a_2 \leq \cdots \leq a_n.$$

We denote by $\sigma(L)$ the optimum value of problem (PP) and by $\hat{\sigma}(L)$ the optimum value of problem (IPP) from section 2 and let

$$\text{gap}(L) = \sigma(L) - \hat{\sigma}(L)$$

denote the corresponding duality gap.

LEMMA 4.1. *Let $\epsilon > 0$ be such that $\epsilon^{-1} \in \mathbb{N}$. Then $k \geq \epsilon n$ implies*

$$\text{gap}(L) \leq 2\epsilon k + \epsilon^{-2}.$$

Proof. Let $m = \epsilon^{-2}$ and $h = \lfloor n/m \rfloor$ and note $h \leq \epsilon^2 n \leq \epsilon k$.

Divide L into $m+1$ consecutive sublists $L = L_1, \ldots, L_m R$ such $|L_i| = h$, $i = 1, \ldots, m$, and $|R| \leq m$. Denote by a_{i_j} the first (and hence smallest) element of L_j, i.e., $a_{i_1} = a_1$, $a_{i_2} = a_{h+1}$, etc..

We consider the lists $L_j^- = a_{i_j}, \ldots, a_{i_j}$, which arise from the L_j's by replacing each item with a copy of the smallest item in the sublist. This process yields the modified list

$$L^- \;=\; L_1^-, \ldots, L_m^- R.$$

With $L_j^+ = L_{j+1}^-$, $j = 1, \ldots, m-1$, and $L_m^+ = 11, \ldots, 1$, we also consider a related modified list

$$L^+ \;=\; L_1^+, \ldots, L_{m-1}^+ R L_m^+.$$

It is straightforward to see that

(1) $\hat{\sigma}(L^+) \;\geq\; \hat{\sigma}(L^-)$.

Indeed, every (integral) binpacking relative to L^- implies a packing relative to L^+ of at least the same value: empty each bin containing an item of size a_{i_1} and fill it with an item of size 1.

Similarly, any integral binpacking relative to L^+ yields a feasible packing for L if we replace the lth element of L_j^+ by the lth element of L_j. The decrease in value is then bounded by

$$h(a_{i_2} - a_{i_1}) + h(a_{i_3} - a_{i_2}) + \cdots + h(1 - a_{i_m}) \;\leq h \;\leq\; \epsilon k.$$

Together with (1), this shows

(2) $\hat{\sigma}(L) \;\geq\; \hat{\sigma}(L^-) - \epsilon k$.

On the other hand, each feasible fractional binpacking relative to L yields a feasible fractional binpacking relative to L^- if we simply replace the items of L by the corresponding items of L^-. Because $\sum_{I \in \mathcal{F}} \sigma_I y_I = \sum_{i=1}^n a_i \sum_{I \ni i} y_I$, the resulting decrease in value is seen to be at most ϵk. Thus

(3) $\sigma(L^-) \;\geq\; \sigma(L) - \epsilon k$.

Since L^- has at most $2m$ different item sizes, we know from Theorem 3.3 that $\text{gap}(L^-)$ is bounded by m. So we deduce from (2) and (3) that

$$\text{gap}(L) \;\leq\; 2\epsilon k + m. \qquad \square$$

We want to remove the correlation between k and n in Lemma 4.1.

LEMMA 4.2. *Let $\epsilon > 0$ be such that $\epsilon^{-1} \in \mathbb{N}$ and that $\epsilon < a_1 \leq \cdots \leq a_n$ holds for the list L . Then*

$$\textit{gap}(L) \;<\; 2\epsilon^2 k + \epsilon^{-4}.$$

Proof. Consider an optimal basic fractional solution y^* for the linear program (PP). By induction on the number n of items, we may assume that each item i occurs in some feasible set I with $y^*_I \neq 0$. Because each feasible set contains at most $(\epsilon^{-1} - 1)$ items, we obtain the upper bound

$$n \;\leq\; |\mathrm{supp}(y^*)|\,(\epsilon^{-1} - 1)$$

on the number of items.

Note that each item i with $\sum_{I \ni i} y^*_I = 1$ contributes more than ϵ to the objective function value. So there can be no more than k/ϵ such items i.

On the other hand, (PP) is full-dimensional. Thus y^* must satisfy $|\mathcal{F}|$ restrictions with equality. Hence the preceding observation implies the bound $|\mathrm{supp}(y^*)| \;\leq\; 1 + k/\epsilon$.

This shows that $k \geq \eta n$ holds with $\eta = \epsilon^2$. Therefore, Lemma 4.1 yields the bound

$$\mathrm{gap}(L) \;<\; 2\epsilon^2 k + \epsilon^{-4}. \qquad \square$$

THEOREM 4.3. *Let L be an arbitrary list of items and let $\epsilon > 0$ be such that $\epsilon^{-1} \in \mathbb{N}$. Then*

$$\mathit{gap}(L) \;\leq\; \max\{\epsilon k, 2\epsilon^2 k + \epsilon^{-4}\}.$$

Proof. Let L' be the sublist of L containing all items of size larger than ϵ. Furthermore, let y' be an optimal (integral) binpacking relative to L' of value $\hat{\sigma}(L')$. We now try to improve y' by adding as many items from $L \setminus L'$ into the remainig "free" space of the k bins. Denote the resulting (integral) packing by $\hat{y}$. We consider two cases.

Case 1. $\hat{y}$ does not use all items from $L \setminus L'$.

Then each bin is filled by $\hat{y}$ to at least weight $1 - \epsilon$. Hence

$$\mathrm{gap}(L) \;\leq\; k - k(1 - \epsilon) \;\leq\; \epsilon k.$$

Case 2. $\hat{y}$ uses all items from $L \setminus L'$.

Because

$$\sigma(L) \;\leq\; \sigma(L') + \sum_{i \in L \setminus L'} a_i \;\text{ and }\; \hat{\sigma}(L) \;\geq\; \hat{\sigma}(L') + \sum_{i \in L \setminus L'} a_i,$$

we have

$$\mathrm{gap}(L) \;\leq\; \mathrm{gap}(L'),$$

and the bound follows from Lemma 4.2. $\square$

We are now in the position to prove Conjecture 2, employing the same argument that allowed us to derive Conjecture 2 from Conjecture 1 in section 1.

COROLLARY 4.4. *Assume $\epsilon > 0$ is such that $\epsilon^{-1} \in \mathbb{N}$. Let the binpacking game be given by the list L and $k \geq 48\epsilon^{-5}$ bins. Then ϵ-core$(L) \neq \emptyset$.*

Proof. It follows from Woeginger [16] that every binpacking game has a non-empty ϵ-core if $\epsilon \geq 1/3$. So we can assume $\epsilon < 1/3$. In the latter case, we observe

$$2(\epsilon/2)^2 k + (\epsilon/2)^{-4} \;<\; \epsilon k/2.$$

Relative to $\epsilon/2$, Theorem 4.3 therefore implies $\text{gap}(L) \leq \epsilon k/2$.

Now consider an optimal integral packing $\hat{y}$ of the items from the list L into the k bins. If $\hat{y}$ uses all items, then the 0-core(L) is nonempty (a core vector is obtained by assigning to each item i its weight $x_i = a_i$ and 0 to each bin).

If $\hat{y}$ does not use all the items, each bin is filled to at least half of its capacity, i.e., $\hat{\sigma}(L) \geq k/2$. So Lemma 2.1 guarantees that the ϵ-core is nonempty. □

5. Approximating the best ϵ. Given an arbitrary binpacking game via a list L of items and k bins, there exists a minimal $\epsilon(L,k) \geq 0$ such that ϵ-core(L) is nonempty whenever $\epsilon \geq \epsilon(L,k)$. In fact, Lemma 2.1 immediately yields

$$\epsilon(L,k) \;=\; 1 \;-\; \frac{\hat{\sigma}(L)}{\sigma(L)}.$$

In this section, we will show that for every fixed $\delta_0 > 0$, there exists a polynomial algorithm that, on input L and $k > 1/\delta_0$, computes an ϵ^* with the properties

(i) ϵ^*-core(L) is nonempty;

(ii) $\epsilon^* - \epsilon(L,k) \;\leq\; \delta_0$.

Our algorithm is motivated by the theoretical analysis of the preceding section. We will use the following notation. We define

$$\begin{array}{rcl} k_0 & := & \min\{k \,|\, k > 1/\delta_0\}, \\ \delta & := & \lceil (\delta_0 - 1/k_0)^{-1} \rceil^{-1}. \end{array}$$

L_δ is the sublist of L which contains all items of size larger than δ. (IPP_δ) and (PP_δ) are the optimization problems (IPP) and (PP) relative to the list L_δ.

If $k \geq 48\,\delta^{-5}$, we know from Corollary 4.4 that δ-core(L) is nonempty and hence $0 \leq \epsilon(L,k) \leq \delta$ allows us to take $\epsilon^* = \delta$.

We can therefore assume that the number k of bins is bounded by the constant $C = C(\delta) = 48\,\delta^{-5}$. Because no feasible set of items relative to L_δ contains more than $1/\delta$ items, there are at most the polynomially bounded number

$$n^{k/\delta}$$

of feasible solutions for (IPP_δ). This means that we can find an optimal solution $\hat{y}_\delta$ for (IPP_δ) by enumeration in polynomial time. We now try to improve $\hat{y}_\delta$ by adding as many items from $L \setminus L_\delta$ as possible greedily into the remaining space of the bins and obtain the feasible solution $\hat{y}$ for (IPP).

If $\hat{y}$ does not use all the items from $L \setminus L_\delta$, we know from

$$\hat{\sigma}(L) \geq \sigma^T \hat{y} \geq (1-\delta)k$$

that δ-core(L) $\neq \emptyset$, i.e., $\epsilon^* = \delta$ suffices.

For the remainder, we thus assume that $\hat{y}$ does use all the items from $L \setminus L_\delta$ and note that, consequently, $\hat{y}$ must be an optimal solution for (IPP) (otherwise, $\hat{y}_\delta$ could not have been optimal for (IPP_δ)).

Compute an optimal solution y^δ for the linear program (PP_δ). Since the number of restrictions of (PP_δ) is polynomially bounded, y^δ can be found in polynomial time. We try to improve y^δ, making use of items in $L \setminus L_\delta$ so that the resulting vector y is feasible for (PP). Again, we proceed greedily as follows.

Find feasible sets $I_1, \ldots, I_r$ such that $y^\delta_{I_i} > 0$, and the following three properties hold:

(1) $\sigma(I_i) \le 1-\delta$;

(2) $\sum_{i=1}^{r-1} y^\delta_{I_i} < 1$;

(3) $\sum_{i=1}^{r} y^\delta_{I_i} \ge 1$.

Choose $a \in L \setminus L_\delta$ of size $s(a)$ and define the vector $\tilde{y}$ via

$$\tilde{y}_I = \begin{cases} 0 & \text{for } i = 1, \dots, r-1, \\ y^\delta_{I_i} & \text{if } I = I_i \cup a \text{ and } i \le r-1, \\ 1 - \sum_{i=1}^{r-1} y^\delta_{I_i} & \text{if } I = I_r \cup a, \\ y^\delta_{I_r} - \tilde{y}_{\{I_r \cup a\}} & \text{if } I = I_r, \\ y^\delta_I & \text{otherwise.} \end{cases}$$

Then $\tilde{y}$ is feasible for (PP) and $\sigma^T \tilde{y} = \sigma^T y^\delta + s(a)$.

Repeat the algorithm with $\tilde{y}$ instead of y^δ and stop with the vector y if no further iteration is possible. There are two possibilities.

Case 1. All elements of $L \setminus L_\delta$ have been used.

Then y must indeed be an optimal solution for (PP). Hence we have

$$\epsilon(L,k) \;=\; 1 \;-\; \frac{\sigma^T \hat{y}}{\sigma^T y}.$$

Case 2. Not all elements of $L \setminus L_\delta$ have been used.

Then it is not hard to see that $\sigma^T y \ge (1-\delta)(k-1)$ must hold. Again, we consider two cases.

If $\hat{\sigma}(L) \ge (1-\delta)(k-1)$, then $1 - (\hat{\sigma}(L)/\sigma(L)) \;\le\; \delta + 1/k$, i.e., we may take $\epsilon^* = \delta_0$.

Generally, we observe

$$1 - \frac{\sigma^T \hat{y}}{(1-\delta)(k-1)} \;\le\; \epsilon(L,k) \;\le\; 1 - \frac{\sigma^T \hat{y}}{k}.$$

If $\hat{\sigma}(L) < (1-\delta)(k-1)$, then

$$\hat{\sigma}(L)\left(\frac{1}{(1-\delta)(k-1)} - \frac{1}{k}\right) \;<\; 1 - \frac{(1-\delta)(k-1)}{k} \;\le\; \delta + \frac{1}{k} \;\le\; \delta_0.$$

Since $k \ge k_0$, $(1 - \hat{\sigma}(L)(1-\delta)^{-1}(k-1)^{-1})$ and $(1 - \hat{\sigma}(L)/k)$ then determine a confidence interval for $\epsilon(L,k)$ of length at most δ_0.

We summarize the results in this section.

THEOREM 5.1. *For every fixed $\delta > 0$, there is a polynomial algorithm that, on input of the list L and the number $k > 1/\delta$ of bins, determines an interval containing $\epsilon(L,k)$ and having length at most δ.*

6. Remarks and open problems. Several questions regarding computational complexity arise naturally in the context of binpacking games (or, more generally, in "combinatorial games"). To wit: Given an instance of a binpacking game,

(1) Is the (ϵ-)core nonempty?

(2) Is a given vector x an (ϵ-)core allocation?

While there are many similarities with respect to the usual complexity aspects of combinatorial optimization problems, noteworthy discrepancies arise from the gametheoretic model (see also Deng and Papadimitriou [1]). For example, the integer program (IPP) is NP-hard. For $k = 2$ and $\sum_i a_i = 2$, it is equivalent with PARTITION (cf. Garey and Johnson [6]). But even for $k = 1$, no polynomial algorithm to solve (IPP) exists unless $P = NP$ holds. Yet, the answer to question (1) above is trivially "yes" when $k = 1$: one simply allocates the optimal value of (IPP) (whatever it might be) to the bin player.

On the other hand, the latter observation implies that testing membership in the feasibility region of the allocation problem (AP) in section 2 is, already for $k = 1$, at least as hard as solving a knapsack problem. Indeed, the allocation vector assigning value q to the bin and value 0 to any other item is feasible for (AP) if and only if q is an upper bound on the total size of items that can be packed into the "knapsack."

Moreover, Woeginger [17] noted that it is hard to decide whether a given allocation vector lies in the core of a binpacking game already in the case $k = 1$ (unless $P = NP$). His construction relates a binpacking game with one bin to the NP-complete problem SUBSET SUM as follows.

Let integers $q_1, \ldots, q_n$, and r, $q_i \leq r$, be given. Does there exist a subset of the q_i's with sum exactly r?

Define a binpacking game with 1 bin and $n + 1$ items $0, 1, \ldots, n$ of sizes $a_0 = 1$ and $a_i = q_i/r$ $(i = 1, \ldots, n)$. Consider an allocation x which assigns value $(1 - r^{-1})$ to the bin, value r^{-1} to item 0, and value 0 to each of the other items. Then x lies in the core of the binpacking game if and only if no subset of the integers q_i has sum exactly r.

We suspect that question (1) above is hard, too. But we are not aware of a strict proof of this conjecture.

We also leave it as an open problem to decide whether our Theorem 5.1 can be improved in the sense that the assumption "$k > 1/\delta$" can be dropped.

REFERENCES

[1] X. Deng and C.H. Papadimitriou, *On the complexity of cooperative solution concepts*, Math. Oper. Res., 19 (1994), pp. 257–266.

[2] U. Faigle, *Cores of games with restricted cooperation*, ZOR—Math. Methods Oper. Res., 33 (1989), pp. 405–422.

[3] U. Faigle and W. Kern, *On some approximately balanced combinatorial cooperative games*, ZOR—Math. Methods Oper. Res., 38 (1993), pp. 141–152.

[4] U. Faigle and W. Kern, *On the core of ordered submodular cost games*, Math. Programming, to appear.

[5] U. Faigle, S.P. Fekete, W. Hochstättler, and W. Kern, *The nucleon of cooperative games and an algorithm for matching games*, Math. Programming, to appear.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, 1979.

[7] J. Kuipers, *Bin packing games*, Zeit. Oper. Res., to appear.

[8] L. Lovász and M.D. Plummer, *Matching Theory*, North–Holland Math. Stud. 121, North-Holland, Amsterdam, 1986.

[9] A. van den Nouweland, J. Potters, S. Tijs, and J. Zarzuelo, *Cores and Related Solution Concepts for Multi-choice Games*, Res. Memorandum FEW 478, University of Tilburg, 1991.

[10] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1944, 1947.

[11] L.S. Shapley, *On balanced sets and cores*, Naval Res. Logistics Quart., 14 (1967), pp. 453–460.

[12] L.S. Shapley, *Cores and convex games*, Internat. J. Game Theory, 1 (1971), pp. 1–26.

[13] S. Tijs and P. Borm, *Operations research, games and graphs*. ZOR—Math. Methods Oper. Res., 38 (1993), pp. 109–110.
[14] F. de la Vega and G.S. Lueker, *Bin packing can be solved within* $1 + \epsilon$ *in linear time*, Combinatorica, 1 (1981), pp. 349–356.
[15] R.J. Weber, *Games in coalitional form*, in Handbook of Game Theory II, R.J. Aumann and S. Hart, eds., North–Holland, Amsterdam, 1994, pp. 1285-1303.
[16] G.J. Woeginger, *On the rate of taxation in a cooperative bin packing game*, ZOR—Math. Methods Oper. Res., 42 (1995), pp. 313–324.
[17] G.J. Woeginger, *Private communication*, 1995.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 400–413, August 1998

005

TIME AND COST TRADE-OFFS IN GOSSIPING*

ARTUR CZUMAJ†, LESZEK GĄSIENIEC‡, AND ANDRZEJ PELC§

Abstract. Each of n processors has a value which should be transmitted to all other processors. This fundamental communication task is called *gossiping*. In a unit of time every processor can communicate with at most one other processor and during such a transmission each member of a communicating pair learns all values currently known to the other.

Two important criteria of efficiency of a gossiping algorithm are its running time and the total number of transmissions. Another measure of quality of a gossiping algorithm is the total number of links used for transmissions. This is the minimum cost of a network which can support the gossiping algorithm. We establish trade-offs between the time T of gossiping and the number C of transmissions and between the time of gossiping and the number L of links used by the algorithm.

For a given T we construct gossiping algorithms working in time T, with parameters C and L close to optimal.

Key words. algorithm, lower bounds, gossiping, time, cost

AMS subject classifications. 05C85, 94C15, 68Q22, 68R10

PII. S0895480295292934

1. Introduction.

Gossiping (also called all-to-all broadcasting) is one of the fundamental tasks in network communication. Every node of a network (processor) has a piece of information (value) which has to be transmitted to all other nodes by exchanging messages along the links of the network. Gossiping algorithms have been extensively studied, especially in the last twenty years; see the comprehensive surveys [5, 8] of the domain.

The classical communication model, already used in the early papers on gossiping [1, 2, 3, 7, 14], is called the 1-*port full-duplex* model. Communication is synchronous. In a single round (lasting one unit of time) every node can communicate with at most one neighbor, and during such a transmission communicating nodes exchange all the values they currently know.

Two important criteria of efficiency of a gossiping algorithm are its *running time* (the number of communication rounds) and the total number of transmissions (calls). The latter is a measure of cost of the algorithm, assuming unit charge per call. The minimum time of gossiping in a complete n-node network was the first problem in this domain, studied in the 1950s [2, 14]. It was proved to be $\lceil \log n \rceil$ for even n and $\lceil \log n \rceil + 1$ for odd n. On the other hand, the minimum number of calls in gossiping is $2n - 4$ for any $n > 3$ (cf. [1, 7]).

Another measure of quality of a gossiping algorithm is the total number of links used for communication. This is the minimum cost of a network which can support the

* Received by the editors October 9, 1995; accepted for publication September 15, 1997; published electronically July 7, 1998.

http://www.siam.org/journals/sidma/11-3/29293.html

† Heinz Nixdorf Institute, University of Paderborn, D-33095 Paderborn, Germany (artur@hni.uni-paderborn.de). The research of this author was partially supported by DFG-Graduiertenkolleg "Parallele Rechnernetzwerke in der Produktionstechnik," ME 872/4-1.

‡ Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02-097 Warszawa, Poland (lechu@mimuw.edu.pl). This work was done during the author's stay at the Université du Québec à Hull as a postdoctoral fellow.

§ Département d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada (pelc@uqah.uquebec.ca). The research of this author was partially supported by NSERC grant OGP 0008136.

algorithm, measured by the number of links in the network. This can also be viewed as a measure of the cost of implementing the algorithm, a fixed cost associated with network design, rather than the cost associated with each run. Clearly, the sparsest network supporting gossiping is a tree, and thus the minimum number of links is $n-1$.

It turns out that the above criteria of efficiency are incompatible: it is impossible to minimize time and the number of calls or to minimize time and the number of links used by the algorithm, simultaneously. If $n = 2^r$, every gossiping algorithm working in time r must have both the number of calls and the number of links used for communication equal to $r2^{r-1}$, as every node has to communicate in every round with a different node in order to double its knowledge. On the other hand, Labahn [11] proved that the minimum running time of a gossiping algorithm with the number of calls $2n-4$ is $2\lceil \log n \rceil - 3$, almost a double of the absolute minimum time. (An earlier proof of this fact, published in [15], was incorrect.) Likewise, in order to minimize the number of links used for communication, we must allow larger gossiping time. Labahn [10] proved that the minimum gossiping time in a tree is at least $2\lceil \log n \rceil - 1$, again almost a double of the absolute minimum time.

These results indicate the existence of time vs. cost trade-offs in gossiping, where cost is measured either by the number C of calls or by the number L of links used for communication. Establishing these trade-offs is the main goal of the present paper. For a given T (ranging from $\log n$ to $2\log n$) we show upper and lower bounds on the minimum cost of gossiping in time T. The algorithms yielding our upper bounds are generalizations of known gossiping schemes that minimized separately either the running time or the cost (cf. [1, 2, 3, 7, 11, 13, 14, 15]). While these classical algorithms were either fast but costly or cheap but slow, it turns out that they can be combined to yield almost optimal cost for any given running time. However, the main contribution of this paper is lower bounds on the minimum cost of gossiping for a given running time that closely matches the performance of our respective algorithms. This is the first time that the full spectrum of relations between the time and the cost of gossiping is investigated. This way of stating the problem significantly increases its complexity, as compared to the classical approach concentrating only on extremal parameter values.

The main technical difficulties of this work lie in establishing the lower bounds on the number of messages and the number of links used by a gossiping scheme with given execution time. To do so, we need to control the amount of knowledge gained by all nodes in any information exchange process whose running time is within the imposed bound. This is particularly hard in the case of the lower bound on the number of links for very fast gossiping schemes (see Theorem 5.1).

Each of our bounds is useful for a different range of values of the running time and cost. If the running time is $T = \lceil \log n \rceil + t(n)$, we show an upper bound $2n + O(\frac{n \log n}{2^{t(n)}})$ on the number of calls, which closely matches the lower bound $\Omega(\frac{n \log n}{2^{t(n)}})$ following from [12]. These bounds are useful for small $t(n)$, i.e., when the running time is small. If the running time is $T = 2\lceil \log n \rceil - r(n)$, we show an upper bound $2n + O(r(n)2^{r(n)})$ and a lower bound $2n + \Omega(\frac{2^{r(n)}}{\log^2 n})$. These bounds are useful for small $r(n)$, i.e., when the running time is larger.

Here are a few consequences of the above results. Let the running time T of gossiping be equal to $\lceil \log n \rceil + t(n)$. Let C denote the minimum number of calls in time T. The following sequence of bounds shows how C gradually decreases from $\Theta(n \log n)$ to the asymptotically optimal range $2n + o(n)$, as restrictions on T are being relaxed.

If $t(n)$ is constant then $C \in \Theta(n \log n)$.
If $t(n) = \log\log n - f(n)$, where $f(n) \to \infty$, then $C \in \omega(n)$.
If $t(n) \geq \log\log n - d$ for a constant d, then $C \in O(n)$.
If $t(n) = \log\log n + f(n)$, where $f(n) \to \infty$, then $C \in 2n + o(n)$.

For medium range values of the running time T we obtain the following bounds on the minimum number of calls:

If $t(n) = \alpha \log n$, where $0 < \alpha < 1$, then $C \in 2n + O(n^{1-\alpha} \log n)$ and $C \in 2n + \Omega(\frac{n^{1-\alpha}}{\log^2 n})$.

Finally, if we want to keep the number of calls very small, time has to increase significantly.

If $C = 2n + c(n)$, where $c(n)$ is polylogarithmic in n, then $T \in 2\log n - o(\log n)$.

We also establish trade-offs between the time T of gossiping and the minimum number L of links used for communication. For medium and large values of T the optimum values of L are roughly one-half of the values of C for the same time. In this range we get bounds that are even tighter than in the case of the number C of calls. For example, if $T = \log n + \alpha \log n$, where $0 < \alpha < 1$, then $L \in n + O(n^{1-\alpha} \log n)$ and $L \in n + \Omega(\frac{n^{1-\alpha}}{\log n})$. For small values of $T = \log n + t(n)$ we obtain the upper bound $n + O(\frac{n \log n}{2^{t(n)}})$ on L, but our lower bound leaves a larger gap than before: we show that if $t(n) \leq c \log\log\log n$ for $c < 1$, then $L \in \omega(n(\log\log n)^d)$, for $d < 1 - c$. It remains open, for example, if $L \in \Omega(n \log n)$ for constant $t(n)$.

The latter bound should be contrasted with a result of Grigni and Peleg [6], concerning broadcasting. They showed that the minimum number of links in an n-node network supporting broadcasting from any node in a given time T is extremely sensitive to the value of T: if n is a power of 2, broadcasting in time $\log n$ requires $\Omega(n \log n)$ links, while broadcasting in time $\log n + 1$ can be performed in a network with $O(n)$ links. Our bound shows that this is not the case for gossiping: in particular, gossiping in time $\log n + \text{const}$ cannot be performed in a network with a linear number of links.

It turns out that the problem of minimizing the cost of gossiping with a given running time has a different flavor in the case of the number of calls and of the number of links. While the same algorithms provide upper bounds in both cases, the techniques used to prove lower bounds are different, and results concerning one of these performance measures do not seem to imply meaningful bounds for the other, in any straightforward way.

Our positive results permit us to choose a gossiping scheme which uses the right balance of resources (time, number of messages, number of links) in a given application. The significance of our negative results lies in a more realistic assessment of feasibility of fast gossiping schemes. While very fast schemes are theoretically possible, their high cost may make them inapplicable in practice: an excessive number of messages (high values of parameter C) is likely to cause network congestion, while dense networks (high values of parameter L) are usually difficult to implement. In situations where these drawbacks are prohibitive, our lower bounds may suggest settling for a slower scheme.

The paper is organized as follows. In section 2 we introduce the terminology and state some preliminary results used in what follows. Section 3 is devoted to the description of a class of gossiping algorithms and computing their running time, number of calls, and number of links used for communication. These results yield upper bounds on the minimum cost of gossiping with a given running time. In section 4 we establish lower bounds on the number of calls in gossiping with a given running

time. In section 5 we give lower bounds on the number of links used in gossiping with a given running time. In section 6 we derive consequences of previous results by applying them with appropriate parameter values. Finally, section 6 contains conclusions and open problems.

2. Terminology and preliminaries. The set of communicating nodes is denoted by X and its size is denoted by n. A *calling scheme* S on the set X is a multigraph on X whose edges are labeled with natural numbers $1, ..., t$ so that edges sharing a common node have different labels. Edges with label i represent calls made in the ith time unit. The number of labels is called the *running time* of the scheme and the number of edges is called the *number of calls* of the scheme. The corresponding multigraph is called the *graph of calls* of the scheme S. The *underlying graph* of a calling scheme S is the simple graph on the set X of nodes in which adjacent nodes are those joined by at least one edge in S. This is the minimal network that supports the scheme S. The number of edges in the underlying graph of S is called the *number of links* used by S.

Upon completion of S the node v knows the value of the node w if there exists an *ascending path* from w to v in S, i.e., a path with increasing labels on edges. The set of nodes who know the value of v upon completion of S is denoted by $K(v)$, and the set of nodes whose values are known to v upon completion of S is denoted by $K^-(v)$. If $K(v) = K^-(v) = X$ for all $v \in X$, the calling scheme S is called a *gossiping scheme* or *gossiping algorithm*. The (total) *knowledge* upon completion of the calling scheme S is the number $K = \sum_{v \in X} |K^-(v)|$. The knowledge after i rounds is at most $n2^i$, because $|K^-(v)| \leq 2^i$ for every node v. The knowledge at the end of a gossiping scheme is n^2.

LEMMA 2.1.

1. *If the calling scheme has k calls then $|K(v)| \leq k+1$ and $|K^-(v)| \leq k+1$ for every node v.*

2. *If the running time of a calling scheme is t then $|K(v)| \leq 2^t$ and $|K^-(v)| \leq 2^t$ for every node v.*

Proof. The proof is straightforward. □

LEMMA 2.2. *If $|K(v)| = k$ then the time required for the remaining $n-k$ nodes to learn the value of v is at least $\log n - \log k$.*

Proof. One of the k informed nodes has to inform at least $\frac{n-k}{k}$ other nodes which requires time at least $\log \frac{n}{k} = \log n - \log k$. □

All logarithms are with base 2. The notation O, Ω, and Θ is standard. We use $o(f(n))$ (resp., $\omega(f(n))$) to denote the class of functions $g(n)$ such that $\frac{g(n)}{f(n)}$ (resp., $\frac{f(n)}{g(n)}$) converges to 0, as n grows.

3. Gossiping algorithms and upper bounds. In this section we present a class of gossiping algorithms that provide good time and cost trade-offs in both the case when cost is measured by the number of calls and when it is measured by the number of links used for communication. Two important graphs will be used in the construction of our schemes. The first is the k-dimensional *hypercube* H_k. This is the graph on 2^k nodes labeled with all binary sequences of length k. Nodes are adjacent iff their labels differ in exactly one position. Nodes whose labels differ in the jth position are called *j-neighbors*.

The second graph is the *k-broadcasting tree* B_k. It is defined by induction on k. B_0 is a single node v. B_{k+1} is obtained from B_k by attaching a different new node to

every node of B_k. The set of all new edges is called the $(k+1)$th *layer* in B_{k+1}. The initial node v is called the *root* of the broadcasting tree.

Hypercubes and broadcasting trees are important for gossiping. Giving the label j to edges of the hypercube H_k joining j-neighbors yields a gossiping scheme with the smallest running time k. The cost of this scheme, however, is very large: it uses $k2^{k-1}$ calls and $k2^{k-1}$ links. On the other hand, broadcasting trees yield gossiping schemes with small cost but large time. Replace every edge in layers $j = 2, ..., k$ of B_k by two edges: one with label $k-(j-1)$ and the other with label $k+(j-1)$. Give label k to the edge in layer 1. The obtained gossiping scheme first gathers all values in the root and then broadcasts all of them to all nodes. Its running time is $2k-1$, but its cost is very low: if $n = 2^k$ is the number of nodes, it has the optimal number $n-1$ of links and it uses $2n-3$ calls, only one call more than the absolute minimum.

In order to save gossiping time at a given cost or to lower cost with a given running time, it is advantageous to use a combination of the two above schemes. Let $n = 2^k + x$, where $0 < x \le 2^k$. Thus $k = \lceil \log n \rceil - 1$. Let $r \le k$ and $s = k - r$. We describe the gossiping algorithm $\mathrm{COT}(n, r)$. (COT stands for cube of trees.) Consider the hypercube H_r and let each of its nodes be the root of a broadcasting tree B_s. Trees rooted at distinct nodes of H_r are disjoint. There are 2^k nodes in all trees. Attach each of the remaining x nodes to a distinct node in one of the trees. Define the set of edges incident to these nodes to be the $(s+1)$th layer. Replace each edge of layers $j = 1, ..., s+1$ by two edges with labels $s+2-j$ and $s+r+1+j$. Finally, give label $s+1+i$, for $i = 1, ..., r$, to edges of the hypercube H_r joining i-neighbors.

The above described gossiping scheme works as follows: first information from all nodes of the tree rooted at a given node of the hypercube is gathered in this node. Then gossiping is executed inside the hypercube H_r among all its nodes. At this point all nodes of the hypercube know all values. Finally, each node of the hypercube broadcasts the complete information to all nodes of the tree rooted at it. The underlying graph of the scheme $\mathrm{COT}(n, r)$ is the undirected version of the graph $H_{r,s}$ used in [6] for broadcasting.

THEOREM 3.1. *The gossiping algorithm $COT(n, r)$ has running time $T = 2\lceil \log n \rceil - r$ and uses $C = 2n + (r-4)2^{r-1}$ calls and $L = n + (r-2)2^{r-1}$ links.*

Proof. Gathering information in nodes of H_r takes time $s+1$, gossiping in H_r takes time r, and broadcasting complete information in trees takes time $s+1$, for a total of

$$T = r + 2(s+1) = k + s + 2 = 2\lceil \log n \rceil - r.$$

Gathering information in nodes of H_r uses $2^r(2^s - 1) + x$ calls, gossiping in H_r uses $r2^{r-1}$ calls, and broadcasting complete information in trees again uses $2^r(2^s - 1) + x$ calls, for a total of

$$C = r2^{r-1} + 2(2^r(2^s - 1) + x) = r2^{r-1} + 2 \cdot 2^k - 2^{r+1} + 2x = 2n + (r-4)2^{r-1}.$$

The number of links in the hypercube H_r is $r2^{r-1}$ and the total number of links in all trees is $2^r(2^s - 1) + x$, for a total of

$$L = r2^{r-1} + 2^r(2^s - 1) + x = r2^{r-1} + 2^k - 2^r + x = n + (r-2)2^{r-1}. \qquad \square$$

The above theorem yields upper bounds on the cost of gossiping with a given running time. It will be convenient for our purposes to formulate them in two versions.

COROLLARY 3.2. *For any functions $t, r : N \to N$ such that $t(n), r(n) \le \lceil \log n \rceil$, there exists a gossiping algorithm*

1. *with running time* $T = 2\lceil \log n \rceil - r(n)$, *number of calls* $C \in 2n + O(r(n)2^{r(n)})$, *and using* $L \in n + O(r(n)2^{r(n)})$ *links*;

2. *with running time* $T = \lceil \log n \rceil + t(n)$, *number of calls* $C \in 2n + O(\frac{n \log n}{2^{t(n)}})$, *and using* $L \in n + O(\frac{n \log n}{2^{t(n)}})$ *links.*

Proof.

1. The proof is straightforward.

2. Use part 1 for $r(n) = \lceil \log n \rceil - t(n)$. □

The above corollary shows that there exists a gossiping algorithm whose time and cost are both asymptotically optimal, i.e., whose running time is $\log n + o(\log n)$ and which uses $2n + o(n)$ calls and $n + o(n)$ links. To this end it suffices to take, e.g., $t(n) = (\log \log n)^2$. However, the results of the following sections will enable us to establish time and cost trade-offs more precisely.

4. Lower bounds on the number of calls. In this section we give two lower bounds on the number of calls in gossiping with a given running time. Each of them provides meaningful consequences for a different range of time and cost values. The first bound follows directly from a result of Labahn [12] and is useful for small values of the running time.

THEOREM 4.1. *Every gossiping algorithm with running time* $T = \log n + t(n)$ *uses* $C \in \Omega(\frac{n \log n}{2^{t(n)}})$ *calls.*

The next theorem yields lower bounds on the number of calls in gossiping that are useful when the running time is larger. We first prove two lemmas.

LEMMA 4.2. *If a calling scheme has a running time at most* t *and its graph of calls is a tree then*

1. *there exists a node* v *such that* $|K(v)| \leq t + 1$;

2. *there exists a node* v *such that* $|K^-(v)| \leq t + 1$.

Proof. We prove only the first part of the lemma: the second part is analogous. Call a node v *terminal* if there is no ascending path of length larger than 1, starting from v. It suffices to prove that there exists a terminal node v. Indeed, for such a node, $K(v)$ consists of v itself and of its neighbors in the tree of calls. The desired inequality follows from the fact that the number of neighbors of a node in the graph of calls cannot exceed the running time of the calling scheme.

Choose any node w_0 and suppose that it is not terminal. Choose any ascending path (w_0, w_1, w_2) of length 2. If w_2 is terminal, we are done, if not, choose any ascending path (w_2, w_3, w_4) of length 2, and so on. Since labels in each path are strictly increasing and the graph of calls is a tree, at every step at least one new node is visited. Thus the process must terminate at some node w_k which has to be terminal. □

LEMMA 4.3. *If a calling scheme on* n *nodes has a running time at most* t *and uses at most* $n - 1$ *calls then*

1. *there exists a node* v *such that* $|K(v)| \leq t + 1$;

2. *there exists a node* v *such that* $|K^-(v)| \leq t + 1$.

Proof. Again we prove only the first part of the lemma. Suppose that S is a calling scheme satisfying the assumptions but violating assertion 1. Let $a_1, ..., a_k$ be the numbers of nodes in components of the graph of calls of S. No component has a node v such that $|K(v)| \leq t + 1$. It follows from Lemma 4.2 that none of the components can be a tree, hence the ith component must have at least a_i edges. Hence the total number of edges in the graph of calls is at least n, contradicting the assumption on the number of calls in S. □

THEOREM 4.4. *Every gossiping algorithm with running time* $T = 2\log n - r(n)$ *uses* $C \in 2n + \Omega(\frac{2^{r(n)}}{\log^2 n})$ *calls.*

Proof. Let t be the largest integer such that less than n calls are placed before round t. Let S_1^* be the calling scheme consisting of all calls of S with labels at most $t-1$. Lemma 4.3 implies that after time $t-1$ there is a node v such that $|K(v)| \leq 2\log n$. (Here $K(v)$ is taken with respect to the calling scheme S_1^*.) By Lemma 2.2 the additional time required for all nodes to learn the value of v is at least $\log \frac{n}{2\log n} = \log n - \log\log n - 1$. Hence

$$t - 1 + \log n - \log\log n - 1 \leq T,$$

and consequently

$$t \leq \log n + \log\log n + 2 - r(n).$$

Let S_1 be the calling scheme consisting of all calls of S with labels at most t. (The number of calls in S_1 is at least n.) Lemma 2.1 implies that after the first t rounds,

$$|K^-(v)| \leq 2^t \leq \frac{4n\log n}{2^{r(n)}}$$

for every node $v \in X$. (Here sets $K^-(v)$ are taken with respect to the calling scheme S_1.)

Let $a(n) = C - (2n-1)$ and consider the calling scheme S_2 consisting of the first $a(n)$ calls placed after round t (order calls in the same round arbitrarily). Lemma 2.1 implies that, for every node $v \in X$, $|K^-(v)| \leq a(n)+1$, where $K^-(v)$ is taken with respect to S_2. Thus, upon completion of all calls in schemes S_1 and S_2,

$$|K^-(v)| \leq \frac{4n\log n}{2^{r(n)}}(a(n)+1),$$

for every node $v \in X$.

Now at most $n-1$ calls remain to be placed. Denote by S_3 the scheme consisting of these remaining calls. By Lemma 4.3 there exists a node w such that $|K^-(w)| \leq 2\log n$, where now $K^-(w)$ is taken with respect to the scheme S_3. It follows that upon completion of all calls in schemes S_1, S_2, and S_3, i.e., at the end of the scheme S, node w knows the values of at most

$$\frac{4n\log n}{2^{r(n)}} \cdot (a(n)+1) \cdot 2\log n$$

nodes. Since S is a gossiping scheme, we must have

$$\frac{8n\log^2 n(a(n)+1)}{2^{r(n)}} \geq n,$$

from which

$$a(n) \in \Omega\left(\frac{2^{r(n)}}{\log^2 n}\right).$$

This concludes the proof. ∎

5. Lower bounds on the number of links. In this section we establish two lower bounds on the number of links used by a gossiping scheme with a given running time. The first bound concerns the case when the running time is small.

THEOREM 5.1. *Every gossiping algorithm with running time* $T \leq \log n + c \log\log\log n$, *where* $c < 1$ *is a constant, uses* $L \in \omega(n(\log\log n)^d)$ *links, where* $d < 1 - c$.

Before proving the theorem we fix some additional terminology and prove several technical lemmas. Consider a calling scheme with running time T. Let $T = \log n + f(n)$, where $f(n) \leq c \log\log\log n$, $c < 1$. Suppose that the number of links used by this scheme is $L \leq an(\log\log n)^d$ for some constants $a > 0$ and $d < 1 - c$. Then, $L \leq n \cdot 2^{b(n)}$, where $b(n) \leq d^* \log\log\log n$ for $d \leq d^* < 1 - c$ and sufficiently large n. We will prove that the considered calling scheme is not a gossiping scheme. Suppose it is.

A node v is called *weak* after round i of the scheme if $|K^-(v)|$ is at most $\frac{1}{2^{f(n)+2}} 2^i$; a node that is not weak is called *strong*. A call between nodes v and w in round i is said to be α-*increasing* if $|K^-(v)| + |K^-(w)|$ after round i is at most α times larger than $|K^-(v)| + |K^-(w)|$ before this round. Let $\epsilon = \frac{1}{2^{2^{f(n)+b(n)+9}}}$.

In every round i consider the following classes of calls:

A: Calls between weak nodes;

B: $(2 - \epsilon)$-increasing calls not belonging to the class A;

C: All remaining calls.

The idea of the proof is to show that in many rounds there are few nodes that are either weak or participate in calls of class C, and consequently the increase of knowledge in these rounds is too slow to enable achieving knowledge n^2 upon completion of the scheme. Among our arguments many hold only for sufficiently large n. This does not cause any problems, since the result is of asymptotic nature. We skip the phrase "for sufficiently large n" for the sake of brevity.

We start with a lower bound on the number of strong nodes.

LEMMA 5.2. *In every round there are at least* $\frac{3}{2^{f(n)+2}-1} n$ *strong nodes.*

Proof. After every round i the knowledge K is at least $n2^{i-f(n)}$ because in the remaining $\log n + f(n) - i$ rounds knowledge can increase at most $2^{\log n + f(n) - i}$ times and the final knowledge must be n^2. Let p be the number of strong nodes and $n - p$ the number of weak nodes after the ith round. After the ith round the knowledge K is at most $p2^i + (n-p)2^{i-f(n)-2}$, hence

$$p2^i + (n-p)2^{i-f(n)-2} \geq n2^{i-f(n)},$$

which implies

$$p \geq n \cdot \frac{2^{i-f(n)} - 2^{i-f(n)-2}}{2^i - 2^{i-f(n)-2}} = n \cdot \frac{3}{2^{f(n)+2} - 1}. \qquad \square$$

The aim of the next two lemmas is to give an upper bound on the size of the class C. Define the *forbidden distance* to be the maximum number k such that if a call of class C has been placed on a link in round i then no call of this class is placed on this link in rounds $i + 1, \ldots, i + k$.

LEMMA 5.3. *The forbidden distance is at least* $2^{f(n)+b(n)+8}$.

Proof. Suppose that a call of class C has been placed on link $e = (v_1, v_2)$ in round i and let $w = |K^-(v_1)| = |K^-(v_2)|$ be the amount of information in each of these nodes after this round. Let l be the minimum positive integer such that a call of class C is placed on link e in round $i + l$. We will show that $l > 2^{f(n)+b(n)+8}$. Since the call

on link e in round i was in the class C, at least one of the nodes v_1 or v_2 was strong after round $i-1$. Thus

$$w \geq 2^{i-1} \cdot \frac{1}{2^{f(n)+2}} = 2^{i-f(n)-3}. \tag{1}$$

Consider the increase of the number $|K^-(v_1)| + |K^-(v_2)|$ in round $i+l$. Let $w_j = |K^-(v_j)|$ after round $i+l-1$ for $j = 1, 2$. We have

$$w_j \leq w + 2^i + \cdots + 2^{i+l-2} = w + 2^{i+l-1} - 2^i,$$

the upper bound requiring that v_j communicate in every round $i+1, ..., i+l-1$ with nodes having maximum and mutually disjoint information. On the other hand, $|K^-(v_1) \cap K^-(v_2)| \geq w$ after round $i+l-1$ because this inequality was already true after round i.

After round $i+l$ we have

$$w^* = |K^-(v_1)| = |K^-(v_2)| \leq w_1 + w_2 - w;$$

hence the increase of the number $|K^-(v_1)| + |K^-(v_2)|$ in round $i+l$ is at most

$$\frac{2w^*}{w_1+w_2} \leq 2 - \frac{2w}{w_1+w_2} \leq 2 - \frac{2w}{2w + 2^{i+l} - 2^{i+1}} = 2 - \frac{1}{1 + \frac{2^{i+l-1}-2^i}{w}}.$$

In view of inequality (1) the right-hand side of the above is at most

$$2 - \frac{1}{1 + \frac{2^{i+l-1}-2^i}{2^{i-f(n)-3}}} = 2 - \frac{1}{1 + 2^{f(n)+3}(2^{l-1}-1)} \leq 2 - \frac{1}{1+2^{f(n)+l+2}}.$$

Since the call in round $i+l$ on link e is in the class C, it is not in the class B and consequently the number $|K^-(v_1)| + |K^-(v_2)|$ must increase in round $i+l$ more than $2-\epsilon$ times. Hence we get

$$2 - \frac{1}{1+2^{f(n)+l+2}} > 2 - \epsilon,$$

which implies

$$1 + 2^{f(n)+l+2} > 2^{2^{f(n)+b(n)+9}},$$

$$2^{f(n)+l+3} > 2^{2^{f(n)+b(n)+9}},$$

and finally

$$l > 2^{f(n)+b(n)+9} - f(n) - 3 > 2^{f(n)+b(n)+8}. \qquad \square$$

LEMMA 5.4. $|C| \leq \frac{n \log n}{2^{f(n)+7}}$.

Proof. Since the total number of rounds is less than $2\log n$, there are at most $\frac{2\log n}{2^{f(n)+b(n)+8}}$ calls of class C on every link. The total number of links is at most $n2^{b(n)}$; hence

$$|C| \leq n2^{b(n)} \cdot \frac{2\log n}{2^{f(n)+b(n)+8}} \leq \frac{n\log n}{2^{f(n)+7}}. \qquad \square$$

The next two lemmas show that in many rounds there are many strong nodes that do not participate in calls of class C.

Call a round *essential* if there are at most $\frac{n}{2^{f(n)+6}}$ calls of class C in this round.

LEMMA 5.5. *At least $\frac{T}{2}$ rounds are essential.*

Proof. Otherwise, more than $\frac{T}{2}$ rounds would have more than $\frac{n}{2^{f(n)+6}}$ calls of class C, for a total of more than

$$\frac{1}{2}\log n \cdot \frac{n}{2^{f(n)+6}} = \frac{n\log n}{2^{f(n)+7}},$$

which contradicts Lemma 5.4. □

LEMMA 5.6. *In every essential round there are at least $\frac{n}{2^{f(n)+1}}$ strong nodes that do not participate in calls of class C.*

Proof. By Lemma 5.2 there are at most $n(1-\frac{3}{2^{f(n)+2}-1})$ weak nodes in every round. By definition there are at most $\frac{n}{2^{f(n)+6}}$ calls of class C in every essential round. At most $\frac{n}{2^{f(n)+5}}$ nodes can participate in these calls. Hence the total number of nodes that are either weak or participate in a call of class C is at most

$$n\left(1-\left(\frac{3}{2^{f(n)+2}-1}-\frac{1}{2^{f(n)+5}}\right)\right) \le n\left(1-\frac{1}{2^{f(n)+1}}\right),$$

in every essential round. □

The next two lemmas show that in many rounds the rate of knowledge increase can be bounded strictly below 2.

Call a pair of nodes $\{v,w\}$ *red in round i* if $|K^-(v)|+|K^-(w)|$ is at least $\frac{1}{2^{f(n)+2}}2^{i-1}$ after round $i-1$ and if this sum increases at most $2-\epsilon$ times in round i; otherwise, call the pair $\{v,w\}$ *white in round i*.

LEMMA 5.7. *In every essential round there are at least $\frac{n}{2^{f(n)+3}}$ pairwise disjoint red pairs of nodes.*

Proof. Fix an essential round i. A strong node v that does not participate in a call of class C either participates in a call of class B or does not communicate at all in round i. By Lemma 5.6, there are either at least $\frac{n}{2^{f(n)+2}}$ nodes of the first type or of the second type. In the first case there are at least $\frac{n}{2^{f(n)+3}}$ calls in the class B because every such call involves at least one strong node (otherwise it would be in class A). All pairs of nodes in these calls are red, which proves the lemma in this case. In the second case, partition nodes that do not communicate in the ith round into disjoint pairs arbitrarily. Clearly $|K^-(v)|+|K^-(w)|$ does not increase at all in such pairs in the ith round and at least $\frac{n}{2^{f(n)+3}}$ pairs contain a strong node in this case; hence they are red. □

LEMMA 5.8. *In every essential round the total knowledge K increases at most*

$$2-\frac{1}{M^{2^{f(n)}}} \text{ times}, \quad \text{where } M = 2^{2^{b(n)+10}}.$$

Proof. For simplicity assume that the number of nodes is even—it will be clear how to modify the argument otherwise. Fix an essential round i. By Lemma 5.7 there are at least $\frac{n}{2^{f(n)+3}}$ pairwise disjoint red pairs in round i. For every such pair $\{v,w\}$,

$$|K^-(v)|+|K^-(w)| \ge \frac{1}{2^{f(n)+2}}2^{i-1}$$

after round $i-1$, and the increase of $|K^-(v)|+|K^-(w)|$ in this round is at most $2-\epsilon$ times. For pairs $\{v,w\}$ that are white in round i, $|K^-(v)|+|K^-(w)| \le 2^i$ after round $i-1$ and the increase of $|K^-(v)|+|K^-(w)|$ is at most two times.

We want to establish an upper bound on the rate of knowledge increase in round i. We will compute this rate as a fraction R whose numerator is the sum of $|K^-(v)|+|K^-(w)|$ over disjoint pairs of nodes after round i and the denominator is the corresponding sum before round i.

The value of R cannot decrease if the number of red pairs is decreased to $\frac{n}{2^{f(n)+3}}$ and the sum $|K^-(v)|+|K^-(w)|$ after round $i-1$ is lowered to $\frac{1}{2^{f(n)+2}}2^{i-1}$ in every red pair, while the number of white pairs is increased to $\frac{n}{2}-\frac{n}{2^{f(n)+3}}$ and the sum $|K^-(v)|+|K^-(w)|$ after round $i-1$ is increased to 2^i in every white pair. Also, R cannot decrease if we assume that the increase of $|K^-(v)|+|K^-(w)|$ is $2-\epsilon$ times in red pairs and two times in white pairs. Hence we get

$$R \le \frac{\frac{n}{2^{f(n)+3}}(2-\epsilon)\frac{1}{2^{f(n)+2}}2^{i-1}+(\frac{n}{2}-\frac{n}{2^{f(n)+3}})\cdot 2\cdot 2^i}{\frac{n}{2^{f(n)+3}}\frac{1}{2^{f(n)+2}}2^{i-1}+(\frac{n}{2}-\frac{n}{2^{f(n)+3}})\cdot 2^i}.$$

Denote $x=2^{f(n)}$; simplifying gives us

$$R \le \frac{\frac{1}{32x^2}(2-\epsilon)+2(1-\frac{1}{4x})}{\frac{1}{32x^2}+(1-\frac{1}{4x})} = 2-\frac{\epsilon}{32x^2-8x+1} \le 2-\frac{\epsilon}{x^3}$$

and finally

$$R \le 2-\frac{1}{2^{2f(n)+b(n)+9}x^3} \le 2-\frac{1}{M^{2f(n)}},$$

where $M=2^{2^{b(n)+10}}$. □

Proof of Theorem 5.1. Denote, as before, $x=2^{f(n)}$ and $M=2^{2^{b(n)+10}}$. By Lemmas 5.5 and 5.8, knowledge increases at most $2-\frac{1}{M^x}$ times in at least $\frac{1}{2}\log n$ rounds. In all remaining rounds it increases at most 2 times. Hence, in order to show that our scheme is not a gossiping scheme it suffices to show

$$n\left(2-\frac{1}{M^x}\right)^{\frac{1}{2}\log n}\cdot 2^{\frac{1}{2}\log n+f(n)} < n^2,$$

i.e.,

$$\left(1-\frac{1}{2M^x}\right)^{\frac{1}{2}\log n}\cdot 2^{f(n)} < 1. \tag{2}$$

Since $f(n) \le c\log\log\log n$ for $c<1$ and $b(n) \le d^*\log\log\log n$ for $d^*<1-c$, we have

$$4f(n)M^x = 4f(n)2^{2^{f(n)+b(n)+10}} \le \log n.$$

Let $g(n)=2M^x$ and $h(n)=\frac{\log n}{g(n)}$. The latter inequality implies $h(n)\ge 2f(n)$. Since $(1-\frac{1}{g(n)})^{g(n)} \to \frac{1}{e}$, we have $(1-\frac{1}{g(n)})^{g(n)} \le \frac{1}{2.5}$ for sufficiently large n and thus

$$\left(1-\frac{1}{g(n)}\right)^{\frac{1}{2}\log n} = \left(\left(1-\frac{1}{g(n)}\right)^{g(n)}\right)^{\frac{1}{2}h(n)} \le \left(\frac{1}{2.5}\right)^{\frac{1}{2}h(n)}.$$

In view of $h(n) \geq 2f(n)$ we have

$$\left(\frac{1}{2.5}\right)^{\frac{1}{2}h(n)} \cdot 2^{f(n)} < 1,$$

which implies inequality (2). □

The last result of this section gives a meaningful lower bound on the number of links when the running time is in the medium or large range.

THEOREM 5.9. *Every gossiping algorithm with running time* $T \leq 2\log n - r(n)$ *for* $r(n) \geq 0$ *uses* $L \in n + \Omega(\frac{2^{r(n)}}{\log n})$ *links.*

Proof. We may assume that $r(n) = \log\log n + f(n)$, where $f(n) \to \infty$; otherwise the conclusion is trivial. Suppose that $L \leq n + \frac{2^{r(n)}}{16\log n}$. Take a spanning tree of the underlying graph, with root k, diameter at most $2\log n$, and maximum degree at most $2\log n$. Such a tree must exist for the gossiping to be completed in time less than $2\log n$. Color all links of this tree black and all other links (at most $\frac{2^{r(n)}}{16\log n} + 1$ of them) red. Add red links to the tree one by one, each time recoloring red those black links which appear in a newly created cycle. If the link $\{v, w\}$ is added, this causes recoloring red links on the paths joining v with k and w with k in the tree. (Some of them may have been recolored already.) Hence adding a new red link causes recoloring at most $2\log n$ black links. After adding at most $\frac{2^{r(n)}}{16\log n} + 1$ red links, the total number of red links at the end of the recoloring process is at most

$$(2\log n + 1)\left(\frac{2^{r(n)}}{16\log n} + 1\right),$$

which is less than $2^{r(n)-2}$ for sufficiently large n, in view of $r(n) = \log\log n + f(n)$.

Since links that are red at the end of the recoloring process are exactly those situated in cycles in the underlying graph, this graph has $z < 2^{r(n)-2}$ nodes situated in cycles. Hence there exists a tree D attached to only one node d in some cycle such that

$$|D| \geq \frac{n-z}{z} < \frac{n}{2^{r(n)-2}} - 1 > \frac{2n}{2^{r(n)}}.$$

Case 1. $\frac{2n}{2^{r(n)}} < |D| \leq \frac{n}{2}$.

The value of some node v in D reaches the node d after time larger than $\log \frac{2n}{2^{r(n)}} = \log n - r(n) + 1$. Broadcasting the value of v from d to all nodes outside of D requires time at least $\log \frac{n}{2} = \log n - 1$. Hence the total time of gossiping exceeds $2\log n - r(n)$.

Case 2. $|D| > \frac{n}{2}$.

Since the maximum degree of D is at most $2\log n$, the tree D contains a subtree Y such that $\frac{2n}{2^{r(n)}} < |Y| \leq 2\log n \cdot \frac{2n}{2^{r(n)}}$. The rest of the argument is as in Case 1, with D replaced by Y. □

6. Discussion. We have two pairs of bounds on the minimum number of calls C in gossiping with a given running time T. If $T = \lceil\log n\rceil + t(n)$ then $C \in 2n + O(\frac{n\log n}{2^{t(n)}})$ and $C \in \Omega(\frac{n\log n}{2^{t(n)}})$. If $T = 2\lceil\log n\rceil - r(n)$ then $C \in 2n + O(r(n)2^{r(n)})$ and $C \in 2n + \Omega(\frac{2^{r(n)}}{\log^2 n})$. The first pair of bounds is useful for small $t(n)$, e.g., when $t(n) \in O(\log\log n)$, i.e., when gossiping time is small. They yield the following corollary showing how C gradually decreases from $\Theta(n\log n)$ to the asymptotically optimal range $2n + o(n)$, as restrictions on T are being relaxed.

COROLLARY 6.1. *If* $T = \lceil\log n\rceil + t(n)$ *then*

1. *if* $t(n)$ *is constant then* $C \in \Theta(n\log n)$.
2. *if* $t(n) \in \log\log n - \omega(1)$, *then* $C \in \omega(n)$.
3. *if* $t(n) \geq \log\log n - d$ *for a constant* d, *then* $C \in O(n)$.
4. *if* $t(n) \in \log\log n + \omega(1)$, *then* $C \in 2n + o(n)$.

The lower bound $C \in \Omega(\frac{n\log n}{2^{t(n)}})$, following from [12], becomes trivial when $t(n) > \log\log n$. For even larger values of gossiping time our second pair of bounds can be applied. For example, it gives a fairly precise estimate of the minimum number of calls when the running time is in the medium range $\alpha\log n$, where $1 < \alpha < 2$.

COROLLARY 6.2. *If the running time of a gossiping algorithm is* $T = \alpha\log n$, *where* $1 < \alpha < 2$, *then* $C \in 2n + O(n^{2-\alpha}\log n)$ *and* $C \in 2n + \Omega(\frac{n^{2-\alpha}}{\log^2 n})$.

The next corollary corresponds to the situation when the gossiping time is fairly large. In this case it is more natural to reverse the problem: what is the minimum running time of gossiping when the number of calls has to be kept very small?

COROLLARY 6.3. *If the number of calls in a gossiping algorithm is* $C = 2n + c(n)$, *where* $c(n)$ *is polylogarithmic in* n, *then its running time* T *is* $2\log n - o(\log n)$.

Proof. Suppose this is not true, and let $T = 2\log n - r(n)$ for $r(n) \in \Omega(\log n)$. Then $r(n) \geq d\log n$ for some constant d and $C \in 2n + \Omega(\frac{n^d}{\log^2 n})$, a contradiction. □

We next turn our attention to the trade-off between the time T and the number of links L. For small values of T the gap between our upper and lower bounds is larger than in the previous case. Corollary 3.2 and Theorem 5.1 imply, for example, that if $T = \log n + c$, where c is a constant, then $L \in O(n\log n)$ and $L \in \omega(n(\log\log n)^d)$ for $d < 1$. It remains open if $L \in \Omega(n\log n)$ in this case.

The last pair of bounds, applicable for larger values of gossiping time $T = 2\log n - r(n)$, follows from Corollary 3.2 and Theorem 5.9. In this case $L \in n + O(r(n)2^{r(n)})$ and $L \in n + \Omega(\frac{2^{r(n)}}{\log n})$. For the medium range of gossiping time $\alpha\log n$, where $1 < \alpha < 2$, this gives an even more precise estimate of L than was previously given for C.

COROLLARY 6.4. *If the running time of a gossiping algorithm is* $T = \alpha\log n$, *where* $1 < \alpha < 2$, *then* $L \in n + O(n^{2-\alpha}\log n)$ *and* $L \in n + \Omega(\frac{n^{2-\alpha}}{\log n})$.

Finally, a result similar to Corollary 6.3 holds for the number of links.

COROLLARY 6.5. *If the number of links used by a gossiping algorithm is* $L = n + c(n)$, *where* $c(n)$ *is polylogarithmic in* n, *then its running time* T *is* $2\log n - o(\log n)$.

7. Conclusion. We established upper and lower bounds on the minimum number of calls and the minimum number of links used by a gossiping scheme with a given running time. Our algorithms, which turned out to be cost efficient for the whole range of running time values, follow the same simple pattern: gather information in nodes of a hypercube of appropriately chosen size using a separate broadcasting tree for each node, then gossip in the hypercube in minimal time, and finally broadcast complete information to all remaining nodes, again using the same broadcasting trees. The tree part of the scheme uses few calls and few links but a lot of time, as it is executed twice, while the hypercube part is fast but uses many calls and many links. Thus a suitable balance between these parts must be maintained to get low cost for a given running time.

Our bounds leave very small gaps. For example, if $T = \frac{3}{2}\log n$, our upper bound on C is $2n + O(\sqrt{n}\cdot\log n)$ and the lower bound is $2n + \Omega(\frac{\sqrt{n}}{\log^2 n})$, leaving a gap within a factor of $O(\log^3 n)$ in the part of the number of calls exceeding the absolute minimum $2n - 4$. In the case of the number of links L, our bounds are even tighter for this range of running time. For the same value $T = \frac{3}{2}\log n$ as before, our upper bound on L is

$n + O(\sqrt{n} \cdot \log n)$ and the lower bound is $n + \Omega(\frac{\sqrt{n}}{\log n})$, leaving a gap within a factor of $O(\log^2 n)$ in the part of the number of links exceeding the absolute minimum $n - 1$.

Further tightening of these bounds, for all values of running time, remains a natural open problem yielded by our results. We do not know, for example, if it is possible to gossip in time $\frac{3}{2}\log n$ using $2n + \Theta(\sqrt{n})$ calls and/or $n + \Theta(\sqrt{n})$ links. It also remains open what is the minimum value of L when $T = \log n + \text{const}$. We conjecture that $L \in \Theta(n \log n)$ in this case.

Another interesting problem is to evaluate the complexity of finding the exact value of the minimum cost of gossiping with a given running time. Given n and T, can the minimum number of calls C or the minimum number of links L be found in polynomial time?

In many papers (cf. [4, 9, 10]) gossiping was studied for specific important networks, such as trees, grids, or hypercubes, and the time or the number of calls were minimized separately. It would be interesting to extend our study by investigating time vs. number of calls trade-offs in gossiping for these networks as well. Also, communication models other than the classical 1-port full-duplex model (cf., e.g., [9]) could be considered in this context.

REFERENCES

[1] B. Baker and R. Shostak, *Gossips and telephones*, Discrete Math., 2 (1972), pp. 191–193.

[2] A. Bavelas, *Communication patterns in task-oriented groups*, J. Acoust. Soc. Amer., 22 (1950), pp. 725–730.

[3] R.T. Bumby, *A problem with telephones*, SIAM J. Algebraic Discrete Meth., 2 (1981), pp. 13–18.

[4] A. Farley and A. Proskurowski, *Gossiping in grid graphs*, J. Combin. Inform. Systems Sci., 5 (1980), pp. 161–172.

[5] P. Fraigniaud and E. Lazard, *Methods and problems of communication in usual networks*, Discrete Appl. Math., 53 (1994), pp. 79–133.

[6] M. Grigni and D. Peleg, *Tight bounds on minimum broadcast networks*, SIAM J. Discrete Math., 4 (1991), pp. 207–222.

[7] A. Hajnal, E.C. Milner, and E. Szemerédi, *A cure for the telephone disease*, Canad. Math. Bull., 15 (1972), pp. 447–450.

[8] S.M. Hedetniemi, S.T. Hedetniemi, and A.L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks, 18 (1988), pp. 319–349.

[9] D.W. Krumme, *Fast gossiping for the hypercube*, SIAM J. Comput., 21 (1992), pp. 111–139.

[10] R. Labahn, *The telephone problem for trees*, Elektron. Informationsverarb. und Kybernet., 22 (1986), pp. 475–485.

[11] R. Labahn, *Kernels of minimum size gossip schemes*, Discrete Math., 143 (1995), pp. 99–139.

[12] R. Labahn, *Mixed telephone problems*, J. Combin. Math. Combin. Comput., 7 (1990), pp. 33–51.

[13] R. Labahn, *Some minimum gossip graphs*, Networks, 23 (1993), pp. 333–341.

[14] H.G. Landau, *The distribution of completion times for random communication in a task-oriented group*, Bull. Math. Biophys., 16 (1954), pp. 187–201.

[15] J. Nieminen, *Time and call limited telephone problem*, IEEE Trans. Circuits Systems, 34 (1987), pp. 1129–1131.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 414–436, August 1998

006

ON DISTANCE-PRESERVING AND DOMINATION ELIMINATION ORDERINGS*

VICTOR CHEPOI†

Abstract. A *distance-preserving elimination ordering* of a graph G is a linear ordering $v_1, v_2, \ldots, v_n$ of the vertices such that each subgraph $G_i = G(v_1, \ldots, v_i)$, $i < n$, is an isometric subgraph of G. We prove that the ordering of the vertices of a pseudo-modular or a house-free weakly modular graph G produced by the breadth-first search is distance preserving. We specify this result by showing that if, in addition, G does not contain the cycles $C_n, n \geq 5$, and the bipyramids $bipyr(C_m), m \geq 6$, as an isometric subgraph, then any ordering produced by the lexicographic breadth-first search is a *domination elimination ordering* (i.e., every vertex v_i is dominated by some vertex v_j, $j < i$, or, in other words, every vertex v_k, $k < i$, adjacent to v_i is also adjacent to v_j).

Key words. distance-preserving ordering, domination elimination ordering, breadth-first search, lexicographic breadth-first search, weakly modular graph

AMS subject classifications. 05C38, 05C75, 05C85

PII. S0895480195291230

1. Introduction.

Various authors considered specific elimination orderings to characterize certain graph classes or discrete structures. The theory of elimination orderings is used in designing efficient algorithms for solving a number of combinatorial optimization, facility location, and scheduling problems, as well as in Gaussian elimination on sparse systems of linear equations. One of the first classes of graphs to be recognized by a specific ordering of vertices was the class of chordal graphs. They are the graphs having *perfect elimination orderings,* i.e., the greater neighbors of any vertex form a complete subgraph. Linear time algorithms to recognize chordal graphs and to compute a perfect elimination ordering are the *Lexicographic Breadth-First Search (LBFS)* of Rose, Tarjan, and Lueker [35] and the *Maximum Cardinality Search (MCS)* of Tarjan and Yannakakis [39] (Farber and Jamison [22] and Shier [37] prove similar results while studying the notion of induced-path convexity in chordal graphs). Jamison and Olariu [24] introduced the notion of *semisimplicial elimination ordering,* which relaxes that of perfect elimination ordering. They stated that any ordering produced by the procedure LBFS is a semiperfect elimination ordering if and only if the graph does not contain the house, the domino, or any cycle C_n, with $n \geq 5$, as an induced subgraph; such graphs are called HHD-*free* (for a new proof of this result see Dragan, Nicolai, and Brandstädt [19]). Dahlhaus et al. [21] established that if the graph does not contain the house and any cycle C_n, $n \geq 5$ as an induced subgraph (in [21] such graphs are called *HC-free graphs*), then any ordering produced by MCS or LBFS is a domination elimination ordering (a new proof was given by Dragan [20]). Finally, note that Scharlau [36] used a variant of the breadth-first search to prove shellability of a class of numbered simplicial complexes.

*Received by the editors September 6, 1995; accepted for publication (in revised form) June 2, 1997; published electronically July 7, 1998. This research was supported by the Alexander von Humboldt Stiftung and the VW-Stiftung Project No. I/69041 and was performed at the Mathematisches Seminar, Universität Hamburg.

http://www.siam.org/journals/sidma/11-3/29123.html

†Universität Bielefeld, SFB 343, Diskrete Strukturen in der Mathematik, Postfach 100131, D-33501 Bielefeld, Germany (chepoi@mathematik.uni-bielefeld.de). This work was done while the author was on leave from the Universitatea de stat din Moldova, Chişinău.

In recent years several classes of graphs have been investigated from a metric point of view. These are modular graphs (which generalize the median graphs (cf. Bandelt [2], Bandelt and Hedlíková [4], Mulder [30]), distance-hereditary graphs (cf. Bandelt and Mulder [5], Hammer and Maffray [27], Howorka [29]), bridged graphs (which generalize chordal graphs; cf. Anstee and Farber [1], Farber and Jamison [23], Soltan and Chepoi [38]), pseudomodular and pseudomedian graphs (Bandelt and Mulder [6] and Bandelt and Mulder [7]), the absolute retracts of reflexive graphs alias Helly graphs and their bipartite variants (cf. Bandelt and Pesch [10], Nowakowski and Rival [32], Quilliot [33]) and their numerous subclasses. These classes have their distinctive features; however, their members share two metric properties, namely, the triangle and the quadrangle conditions [3, 16, 8]. In [3] such graphs were dubbed *weakly modular graphs*. Again, the recognition problem for a majority of these classes of graphs is solvable in polynomial time by applying special vertex elimination schemes. Usually, the members of these classes admit a *domination elimination ordering,* i.e., an ordering $v_1, \ldots, v_n$ of the vertices such that each v_i is dominated by some $v_j, j < i$, i.e., all vertices $v_k, k < i$, adjacent to v_i are also adjacent to v_j. If, in addition, v_j is a neighbor of v_i, then the graphs admitting such an ordering are called *dismantlable* or *cop-win graphs* in view of the results of Nowakowski and Winkler [31] and Quilliot [34]. That bridged graphs are cop-win graphs has been established by Anstee and Farber [1]. Recently, we proved [18] that a dismantling scheme of a bridged graph can be computed in linear time just by applying the *Breadth-First Search (BFS).* In this note, we continue this line of research by investigating the metric properties of the orderings produced by BFS and LBFS. We prove that for a large class of graphs (house-free weakly modular graphs and all pseudo-modular graphs) every ordering $v_1, \ldots, v_n$ of a graph G generated by BFS is a *distance-preserving ordering,* i.e., every subgraph G_i induced by the vertices $v_1, \ldots, v_i$ is a distance-preserving alias isometric subgraph of G. For this purpose we introduce and investigate the class of pseudopeakless functions (the lower sets of such functions induce isometric subgraphs). To find a distance-preserving ordering of a graph with n vertices is equivalent to constructing a pseudopeakless function which takes n distinct values. As for domination elimination orderings, we characterize the house-free graphs such that any ordering produced by BFS or LBFS of each isometric subgraph is a domination elimination ordering.

2. Weakly modular graphs. All graphs occurring in this note are finite, connected, and without loops or multiple edges. The *distance* $d(u, v)$ between two vertices u and v of a graph G is the length of a shortest path between u and v. The set of all vertices w on shortest paths between u and v is called the *interval* $I(u, v)$ between u and v, that is,

$$I(u, v) = \{x : d(u, v) = d(u, x) + d(x, v)\}.$$

An induced subgraph H of a graph G is *isometric* if the distance between any pair of vertices in H is the same as that in G. An induced subgraph (or a subset of vertices) H is called *convex* if H includes every interval $I(u, v)$ with u, v in H.

The *disk* with center u and radius k is the set

$$D_k(u) = \{x : d(u, x) \leq k\}.$$

We will also use the notation $N[u]$ for the disk $D_1(u)$. In particular, $N[u] = N(u) \cup \{u\}$, where $N(u) = \{x : d(u, x) = 1\}$ is the *(open) neighborhood* of u. More generally,

for a subset S let $N(S)$ denote the *neighborhood* of S, i.e., $N(S) = \cup_{v \in S} N(v)$. Finally, $G(S)$ denotes the subgraph induced by S.

A graph G is *weakly modular* [3, 8, 16] if its shortest-path metric $d = d_G$ satisfies the following two conditions:

Triangle condition: for any three vertices u, v, w with

$$1 = d(v, w) < d(u, v) = d(u, w),$$

there exists a common neighbor x of v and w such that $d(u, x) = d(u, v) - 1$.

Quadrangle condition: for any four vertices u, v, w, z with

$$d(v, z) = d(w, z) = 1 \quad \text{and} \quad d(u, v) = d(u, w) = d(u, z) - 1,$$

there exists a common neighbor x of v and w such that $d(u, x) = d(u, v) - 1$.

We can define weakly modular graphs using the concept of metric triangle. Recall that three vertices u, v, and w form a *metric triangle* uvw if the intervals $I(u, v), I(v, w)$, and $I(w, u)$ pairwise intersect only in the common end vertices. (If uvw is a metric triangle, then any permutation of the letters u, v, and w defines the same metric triangle.) According to [16], G is weakly modular if and only if for every metric triangle uvw all vertices of the interval $I(v, w)$ are at the same distance $k = d(u, v)$ from u. The number k is called the *size* of the metric triangle uvw. A metric triangle uvw is a *pseudomedian* of the triple x, y, z if the following metric equalities are satisfied:

$$d(x, y) = d(x, u) + d(u, v) + d(v, y),$$

$$d(y, z) = d(y, v) + d(v, w) + d(w, z),$$

$$d(z, x) = d(z, w) + d(w, u) + d(u, x).$$

A graph in which every metric triangle is degenerate, that is, has size 0, is called *modular* [2]. In other words, a graph is modular if $I(x, y) \cap I(y, z) \cap I(z, x)$ is nonempty for every triple x, y, z. Now, a *pseudomodular graph* is a graph in which each metric triangle has size at most 1. According to [6], pseudomodular graphs can be characterized in the following way:

If $1 \leq d(u, w) \leq 2$ and $k = d(u, v) = d(v, w) \geq 2$ for vertices u, v, w, then there exists a common neighbor x of u and w such that $d(v, x) = k - 1$.

A graph is called *hereditary weakly modular* if every isometric subgraph is weakly modular. In a similar way we can define *hereditary modular* [2] and *hereditary pseudomodular graphs* [6]. Recall from [2] that hereditary modular graphs are graphs in which all isometric cycles have length four, while hereditary pseudomodular graphs are the graphs which do not contain the house, the 3-sun, or any cycle $C_n, n \geq 5$, as an isometric subgraph (the forbidden isometric subgraphs listed in Figure 2.1). Chordal bipartite and distance-hereditary graphs represent two important instances of hereditary modular and hereditary pseudo-modular graphs, respectively. Finally, hereditary weakly modular graphs are the graphs which do not contain the house or any cycle $C_n, n \geq 5$, as an isometric subgraph [16]. Besides these two classes of graphs, the class of hereditary weakly modular graphs comprises the important classes of *bridged graphs* (where all isometric cycles have length three [23, 38]), HHD-*free graphs* [24], *chordal graphs*, HC-free graphs [21], alias *house-free weakly triangulated graphs* [28].

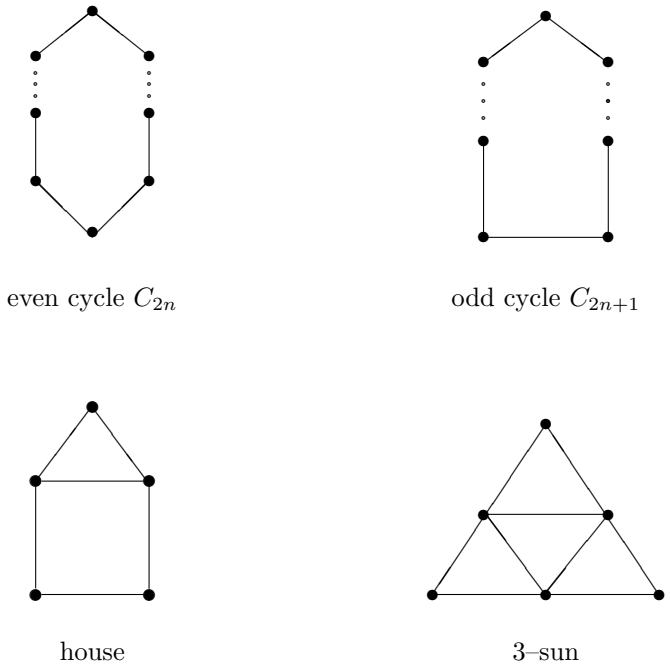


FIG. 2.1. *Forbidden graphs.*

3. Distance-preserving orderings and pseudopeakless functions. For a given linear ordering $v_1, v_2, \ldots, v_n$ of the vertices of a graph G, let G_i denote the subgraph induced by $\{v_1, v_2, \ldots, v_i\}$, and let $N_i(v_j)$ denote $N(v_j) \cap \{v_1, \ldots, v_i\}$, and accordingly $N_i[v_j] = N_i(v_j) \cup \{v_j\}$.

An ordering $v_1, v_2, \ldots, v_n$ of G is *distance-preserving* if each $G_i, i = 1, 2, \ldots, n$ is an isometric subgraph of the graph G. It is quite evident that perfect elimination orderings, semiperfect elimination orderings, and domination elimination orderings are distance-preserving. In Figure 3.1 we present an example of a distance-preserving ordering of a 4-cube.

There are close relationships between distance-preserving orderings and a certain class of functions, which we shall call pseudopeakless. Let $P = (x_0, x_1, \ldots, x_p)$ be a path of a graph G. A real valued function f defined on P is *peakless* if $0 \le j < i < k \le p$ implies $f(x_i) \le \max\{f(x_j), f(x_k)\}$ and equality holds only if $f(x_j) = f(x_k)$. A function f defined on the vertices of a graph G is *peakless* if the restriction of f on any shortest path of G is peakless. Peakless functions were introduced and studied by Busemann [12] and Busemann and Phadke [13] in the geometry of geodesics; see [14] for a recent survey. In graphs, peakless functions were considered in [17]. Peakless functions inherit and generalize the properties of usual convex functions. Now, a function f defined on a graph G is called *pseudopeakless* if any two vertices of G can be joined by a shortest path along which f is peakless. Equivalently, f is pseudopeakless if for any two nonadjacent vertices u, v there is a vertex $w \in I(u, v) - \{u, v\}$ such that $f(w) \le \max\{f(u), f(v)\}$ and equality holds only if $f(u) = f(v)$. The most useful property of pseudopeakless functions is their unimodality, that is, any local minimum of f is global. The proof is simple: let u be a global minimum of f and let v be an

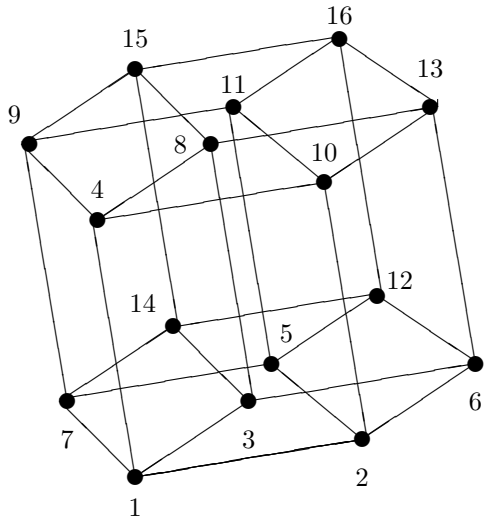


FIG. 3.1. *A distance-preserving ordering of a* 4*-cube.*

arbitrary vertex of G. Consider the shortest path P between u and v along which f is peakless. Then either f is constant on this path, or for the neighbor w of v in P we have $f(w) < \max\{f(u), f(v)\} = f(v)$. In either case, v is also a global minimum; otherwise v is not a local minimum.

Finally, a function f is *strictly pseudopeakless* if any two nonadjacent vertices u, v of G can be connected by a shortest path P along which f is strictly peakless, i.e., $f(w) < \max\{f(u), f(v)\}$ for any $w \in P - \{u, v\}$.

For a subgraph H of a graph G by $f|_H$, we denote the *restriction* of a function f to H. For a real number a let

$$[f \leq a] = \{v \in V : f(v) \leq a\}$$

be the *lower set* of a function f. Let G_a denote the subgraph of G induced by $[f \leq a]$. The following evident lemmas capture some basic properties of pseudopeakless functions.

LEMMA 3.1. *If f is a pseudopeakless function defined on the vertices of a graph G with n vertices, then every G_a is an isometric subgraph of G. Conversely, if f has $|V|$ distinct values and G_a is an isometric subgraph for any a, then f is pseudopeakless.*

LEMMA 3.2. *If $v_1, \ldots, v_n$ is a distance-preserving ordering of G, then the function $\alpha(v_i) = i$ is pseudopeakless. Moreover, G admits a distance-preserving ordering if and only if there is a pseudopeakless function which has n distinct values.*

For similar relationships between perfect elimination orderings and peakless functions, see [37, 17] (more exactly, Shier [37] used quasi-concave and quasi-convex functions, but all quasi-convex functions which have n distinct values are peakless).

LEMMA 3.3. *If H is a convex subgraph of G and f is a pseudopeakless function on G, then $f|_H$ is pseudopeakless on H.*

We shall say that a function g *refines* a function f whenever every lower set of f is a lower set of g. Two functions are *equivalent* if they have identical lower sets.

LEMMA 3.4. *If f is a strictly pseudopeakless function on G, then any refinement g is also strictly pseudopeakless.*

Proof. Let v and w be two nonadjacent vertices of G and $x \in I(v,w) - \{v,w\}$ such that $a = f(x) < \max\{f(v), f(w)\} = f(w)$. Since $[f \le a]$ is a lower set of the function g and $w \notin [f \le a]$, we conclude that $g(x) < g(w)$. □

Let f be a function defined on G. By an *f–minimal path* connecting two vertices u and v, what is meant is a shortest-path $P_f(u,v) = (u = w_0, w_1, \ldots, w_p = v)$ with the minimum sum $\sum_{i=0}^{p} f(w_i)$.

LEMMA 3.5. (1) *A function f on G is pseudopeakless if and only if it is locally pseudopeakless, i.e., for any two vertices u and v at distance two there is a common neighbor w of u and v such that $f(w) \le \max\{f(u), f(v)\}$ and equality holds only if $f(u) = f(v)$.*

(2) *A function f on G is strictly pseudopeakless if and only if for any two vertices u, v, with $d(u,v) = 2$, there exists a common neighbor w of u and v such that $f(w) < \max\{f(u), f(v)\}$.*

Proof. Pick two nonadjacent vertices u and v, and let $P_f(u,v)$ be an f-minimal path connecting u and v. We assert that f is peakless along this path. Since every subpath of $P_f(u,v)$ again is an f-minimal path, it suffices to verify the peaklessness condition only for the vertices $w_0 = u, w_p = v$, and a vertex w_i, where f attains its maximum on $P_f(u,v) - \{u,v\}$. Assume that w_i is as close as possible to w_0. Applying an induction argument on the distance $d(u,v)$ departing from $d(u,v) = 2$, we can assume that f is peakless along each of the f-minimal paths $(w_0, \ldots, w_i)$ and $(w_i, \ldots, w_p)$. Consider the vertices w_{i-1} and w_{i+1}. Since f is locally pseudopeakless, from the choice of the path $P_f(u,v)$ we conclude that $f(w_i) \le \max\{f(w_{i-1}), f(w_{i+1})\}$. From the choice of w_i we obtain that $w_i = w_1$. If $f(w_1) < f(w_0)$, we are done. Otherwise, $f(w_0) = f(w_1) = f(w_2)$. Now, letting w_2 play the role of w_1, and w_1 and w_3 the roles of w_0 and w_2, respectively, we obtain the equality $f(w_3) = f(w_2)$, and so on until we arrive at the vertex w_p. Hence, $f(w_0) = f(w_1) = \cdots = f(w_p)$. □

Similar Tietze-type results were established for convexity in graphs in [9, 16] and for some kind of isometricity of disks of Cayley graphs in [15].

For a graph G, a function f is called *locally unimodal* if the restriction of f on every interval $I(u,v)$ with $d(u,v) = 2$ is a unimodal function on $I(u,v)$.

LEMMA 3.6. *Any locally unimodal function f on G is pseudopeakless. Conversely, if G does not contain the graphs F_1 and F_2 (Figure 3.2) as induced subgraphs, then any pseudopeakless is locally unimodal.*

Proof. Let u and v be two arbitrary vertices of G with $d(u,v) = 2$. If f attains its minimum on $I(u,v)$ in some vertex $w \ne u, v$, then either $f(u) = f(w) = f(v)$ or $f(w) < \max\{f(u), f(v)\}$. So, assume that u is the unique minimum of f on $I(u,v)$. Since v cannot be a local minimum of f, necessarily $f(v) > f(x)$ for some common neighbor x of u and v. Thus, $f(x) < \max\{f(u), f(v)\}$. By Lemma 3.5, f is pseudopeakless.

If G does not contain the graphs F_1 and F_2 as induced subgraphs, then the interval $I(u,v)$ between any two vertices u, v at distance 2 is convex. By Lemma 3.3 the restriction of any pseudopeakless function on $I(u,v)$ is pseudopeakless, and therefore, unimodal. □

The numberings of vertices of the graphs F_1 and F_2 presented in Figure 3.2 provide examples of pseudopeakless but nonlocally unimodal functions.

For a d-polytope $P \subset \mathbb{R}^d$, a numbering of its vertices is called *completely unimodal* [26, 41, 40] if every k-face ($2 \le k \le d$) has a unique local minimum, that is, every face F has only one vertex such that all its neighbors on F get a larger number. In a similar way we can define *completely unimodal functions.* It is established in [26, 41]

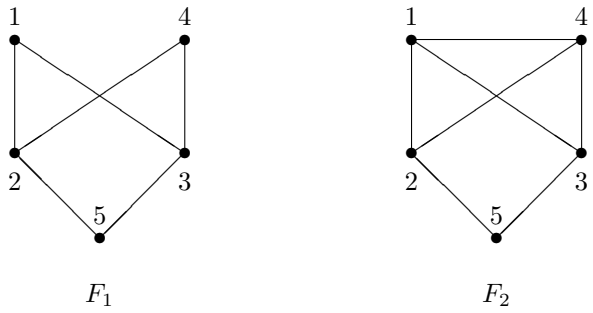


FIG. 3.2.

that if P is the d-dimensional cube, then a numbering is completely unimodal if and only if it is unimodal on every 2-face. Let $G(P)$ denote the graph (1-skeleton) of a polytope P. From previous results we obtain the following fact.

PROPOSITION 3.7. *Let P be a polytope such that all its faces induce convex subgraphs in $G(P)$, and let f be a function defined on the vertices of P.*

(1) *If f is locally unimodal, then f is completely unimodal;*

(2) *If every 2-face of P is a 4-cycle and every $I(u,v)$ with $d(u,v)=2$ constitutes a 2-face of P, then the following conditions are equivalent:*

(i) *f is completely unimodal;*

(ii) *f is unimodal on each 2-face;*

(iii) *f is pseudopeakless.*

The proof of both assertions follows from Lemmas 3.6 and 3.3 and unimodality of pseudopeakless functions.

For a vertex v and a subset of vertices M by $d(v,M)$, we denote the minimum over all distances $d(v,x), x \in M$. For a fixed $M \subseteq V$, let $d_M(v) = d(v,M)$. We shall now characterize the graphs for which the functions $d_u(v) = d(v,u)$ and $d_C(v) = d(v,C)$ are pseudopeakless for all vertices $u \in V$ and all cliques C of G (by a clique we mean any complete subgraph of G). A graph G is called *meshed* [9] if for every triple u,v,w of vertices with $d(v,w)=2$ and

$$1 \le d(u,v) \le d(u,w) \le d(u,v)+1,$$

there exists a common neighbor x of v and w with $d(u,x) \le d(u,v)$.

PROPOSITION 3.8. (1) *For any vertex u of a graph G, the distance function d_u is pseudopeakless if and only if G is meshed. In particular, the distance function d_u of a weakly modular graph is pseudopeakless.*

(2) *for any vertex u of a graph G, the distance function d_u is strictly pseudopeakless if and only if G is pseudomodular.*

The proof of this result is immediate in view of Lemma 3.5. The characterization of pseudomodular graphs as graphs with strictly pseudopeakless functions follows from (2) of Lemma 3.5 and the characterization of pseudomodular graphs of [6] presented in section 2.

PROPOSITION 3.9. *For a graph G the following conditions are equivalent:*

(i) *for any clique C of G, the distance function d_C is pseudopeakless;*

(ii) *for any edge e of G the distance function d_e is pseudopeakless;*

(iii) *G is a weakly modular graph.*

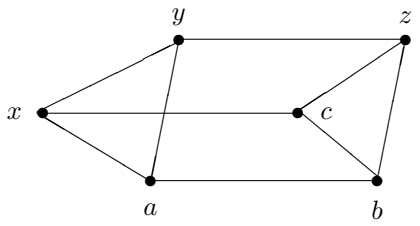


FIG. 3.3.

Proof. (i) → (ii) is evident.

(ii) → (iii). It suffices to verify the quadrangle condition, because any meshed graph satisfies the triangle condition [9]. So, let $p, q \in I(x, y)$ be two neighbors of x, where x and y are selected as close as possible. By Proposition 3.8 there exists a common neighbor w of p and q such that $d(w, y) \leq d(p, y) = d(q, y)$. We can suppose that $d(w, y) = d(p, y)$; otherwise we are done. By the triangle condition there are two common neighbors y' and y'' of p, w and q, w, respectively, one step closer to y. By the induction assumption there exists a common neighbor $z \in I(y', y) \cap I(y'', y)$ of y' and y''. Consider the distance function d_e, where $e = xq$. Then $d_e(p) = 1$, while $d_e(z) = d_e(y') = 2$. Since d_e is pseudopeakless, there is a common neighbor u of p and z with $d_e(u) = 1$. Since $d(x, z) = 3$, the vertex u must be adjacent to q. Therefore, $u \in I(p, y) \cap I(q, y)$, as required.

(iii) → (i). According to Lemma 3.5 it suffices to establish that d_C is locally pseudopeakless. Let $d(u, v) = 2$. Pick two vertices $x, y \in C$ such that $d(u, x) = d(u, C)$ and $d(v, y) = d(v, C)$. We can assume that $d(u, x) < d(u, y)$ and $d(v, y) < d(v, x)$; otherwise u and v will have a common closest vertex in C, and we are in a position to apply Proposition 3.8. Moreover, $|d(u, x) - d(v, y)| \leq 1$; otherwise, if, say $d(u, x) = d(v, y) + 2$, then $d(t, y) = d(v, y) + 1$ for any common neighbor t of u and v, and we are done. We proceed by induction on $d(u, x) + d(v, y)$. If $d(u, x) = d(v, y)$, then $d(v, x) = d(u, x) + 1$. By Proposition 3.8 there is a vertex $w \in I(u, v) - \{u, v\}$ such that $d(w, x) \leq d(u, x)$. So, assume that $d(v, y) = d(u, x) + 1$. Then $u, y \in I(v, x)$. Pick an arbitrary neighbor z of x in $I(v, x)$. By the quadrangle condition we would find a common neighbor t of z and y which is one step closer to v. Consider the clique $C' = \{z, t\}$. By the induction assumption there exists a vertex $w \in I(u, v) - \{u, v\}$ such that $d(w, C') \leq d(u, z)$. Since $d(u, z) = d(u, x) - 1$, this finishes the proof. □

The following example shows that in Proposition 3.8 we cannot replace cliques and edges by isometric subgraphs or shortest paths. In the following graph (which is pseudomodular) consider the isometric subgraph induced by the set $M = \{a, b, c\}$; see Figure 3.3. The function d_M is not pseudopeakless because $d_M(y) = d_M(x) = d_M(z) = 1$; however, $d_M(c) = 0$ and $I(y, c) = \{y, x, z, c\}$.

The results of [17] for graphs, as well as the results for G-spaces and G-surfaces (see [14] for references), show that the nonconstant peakless functions are quite rare. In contrast, at least in meshed or weakly modular graphs nonconstant and even locally nonconstant pseudopeakless functions exist in abundance. Propositions 3.8 and 3.9 provide a method for constructing such kind of functions. Consider for example the distance function d_u. The lower sets of this function are the disks $D_k(u)$ centered at u. In particular, d_u takes the constant value k on the *sphere* $S_k(u) = \{v : d(u, v) = k\}$. One method to produce a distance-preserving ordering of a meshed or a weakly

modular graph G is to extend the function d_u to a pseudo–peakless function which takes $|S_k(u)|$ distinct values on each sphere $S_k(u)$. The simplest way to realize this is to apply the *breadth-first search* (BFS) starting from the vertex u. In the BFS the vertices of a graph G with n vertices are numbered from 1 to n in increasing order. We number with 1 the vertex u and put it on an initially empty queue of vertices. We repeatedly remove the vertex v at the head of the queue and consequently number and place onto the queue all still unnumbered neighbors of v. BFS constructs a spanning tree T_u of G with the vertex u as a root. Then a vertex v is the *father* in T_u of any of its neighbors w in G included in the queue when v is removed. The procedure is called once for each vertex, so the total complexity of its implementation is $O(|V| + |E|)$ (a detailed description of the BFS procedure is presented, for example, in the book of Golumbic [25]). Another method to order the vertices of a graph in linear time is the *lexicographic breadth-first search* (LBFS) proposed by Rose, Tarjan, and Lueker [35]. According to LBFS, the vertices of a graph G are numbered from n to 1 in decreasing order. The *label* $\mathsf{label}(w)$ of an unnumbered vertex w is the list of its numbered neighbors. As the next vertex to be numbered, select the vertex v with the (lexicographic) largest label, breaking ties arbitrarily [35]. Evidently, LBFS is a particular instance of BFS, i.e., every ordering produced by LBFS can also be generated by BFS. Below we reproduce the details of LBFS.

procedure LBFS(G);
Input: the adjacency list of G;
Output: an ordering of the vertices of G.

begin
  for every vertex w in V **do** $\mathsf{label}(w) \leftarrow \emptyset$;
  for $i \leftarrow n$ **downto** 1 **do begin**
    pick an unnumbered vertex v with the largest label;
    number the vertex v with i;
    for each unnumbered $w \in N[v]$ **do**
      add i to $\mathsf{label}(w)$
  end
end;

Let $v_1, \ldots, v_n$ be the ordering of G produced by BFS. For each vertex v let $\alpha(v) = i$ if $v = v_i$. We close the section with some basic properties of the BFS orderings, formulated in terms of the function α. Important convention: in all subsequent results where BFS is used, $\alpha(u) = 1$, i.e., the procedure BFS starts from the vertex u.

LEMMA 3.10. (1) *Let x and y be the fathers in T_u of the vertices v and w, respectively. If $\alpha(v) < \alpha(w)$, then $\alpha(x) \leq \alpha(y)$. Conversely, if $\alpha(x) < \alpha(y)$, then $\alpha(v) < \alpha(w)$;*

(2) *the function α refines the distance function d_u, i.e., if $d(u, v) < d(u, w)$, then $\alpha(v) < \alpha(w)$;*

(3) *the function α is monotone along any shortest path connecting the root u with any vertex v.*

4. Distance-preserving orderings of weakly modular graphs. We are now in a position to state the main result of this paper. We start with a special case of pseudomodular graphs. Already from Lemmas 3.7 and 3.4 we can deduce that α is strictly pseudopeakless, because it refines the strongly pseudopeakless distance function d_u. Alternatively, consider the base point order $<_u$ defined by $v <_u w$ if

and only if $v \in I(u,w)$. Then a function f is called *strictly isotone* if $v <_u w$ implies $f(v) < f(w)$.

LEMMA 4.1. *Any strictly isotone function f on a pseudomodular graph G is strictly pseudopeakless.*

The proof is immediate in view of previous results and because the distance function of pseudomodular graphs is strictly pseudopeakless.

THEOREM 4.2. *Let G be a pseudomodular or a house-free weakly modular graph. Then any ordering $v_1, \ldots, v_n$ of the vertices of G produced by BFS is a distance-preserving ordering.*

Proof. The case of pseudomodular graphs is covered by Lemma 4.1. Henceforth, let G be a house-free weakly modular graph. It is necessary to verify that every G_i, $i = 1, \ldots, n$ is an isometric subgraph of G, or equivalently, that the function α is pseudopeakless. Suppose the contrary, and let G_i be the first nonisometric subgraph among the subgraphs induced by the lower sets of the function α. Pick a vertex w of G_i as close as possible to $v = v_i$ such that v and w cannot be connected inside G_i with a shortest path. Then v and w are the only common vertices of the interval $I(v,w)$ and G_i. Denote by x and y the fathers of the vertices v and w, respectively. First we establish some metric relationships between the vertices v, wx, and y. Let $k = d(u,v)$ and $t = d(v,w)$.

CLAIM 1. *All vertices of the interval $I(v,w)$ are at distance $k = d(u,v)$ from u.*

Proof of Claim 1. Indeed observe that $I(v,w) \cap I(v,u) = \{v\}$ and $I(v,w) \cap I(w,u) = \{w\}$. Otherwise we will get a vertex $p \in I(v,w) - \{v,w\}$ which is closer to u than v or w. By BFS, $\alpha(p) < \max\{\alpha(v), \alpha(w)\} = i$ and $p \in G_i$, contrary to the choice of the vertices v and w. Hence, any pseudomedian of the triple u, v, w has the form u', v, w. From the properties of weakly modular graphs presented in section 2 we know that $d(u', v) = d(u', q) = d(u', w)$ for any vertex $q \in I(v,w)$. Since $u' \in I(u,v) \cap I(u,w)$, we obtain that $d(u,v) = d(u,w) = k \geq d(u,q)$. The choice of the vertices v and w implies that $d(u,q) = k$. This settles Claim 1. □

Claim 1 implies that $x \neq y$. Since $\alpha(w) < \alpha(v)$, we obtain $\alpha(y) < \alpha(x)$ by BFS. From the choice of v and w, we conclude that the function α is peakless along some shortest path $P_\alpha(x,y)$ connecting x and y.

CLAIM 2. *$d(y,v) = d(y,x) = d(v,w) + 1 = d(p,v)$, where p is the neighbor of y in the path $P_\alpha(x,y)$.*

Proof of Claim 2. First we will prove that $d(x,y) \leq d(v,w) + 1$. Consider a shortest-path $P = (v = v_0, v_1, \ldots, v_{t-1}v_t = w)$ between v and w. We infer from Claim 1 that all vertices of P are at distance k from u. Applying the triangle condition to u and the consecutive vertices of the path P, we can find vertices $z_1, \ldots, z_t$ at distance $k-1$ from u such that every z_j is a common neighbor of the vertices v_{j-1} and v_j. Notice that z_j cannot be adjacent to any other vertex of the path P, otherwise $z_j \in I(v,w)$, which is impossible by Claim 1. If x or y are adjacent to at least one vertex v_j, $0 < j < t$, then $d(x,y) \leq d(v,w) + 1$, and we are done. The same inequality holds when the vertices $x = z_0, z_1, \ldots, z_t, z_{t+1} = y$ induce a path. So, we can assume that two consecutive vertices z_{j-1} and z_j are nonadjacent. Since $z_{j-1}, z_j \in I(v_j, u)$, by the quadrangle condition there exists a common neighbor u' of z_{j-1}, z_j at distance $k-2$ to u. Then either the vertices $v_{j-1}, v_j, z_{j-1}, z_j, u'$ or the vertices $v_{j+1}, v_j, z_{j-1}, z_j, u'$ induce a house. This shows that $d(x,y) \leq d(v,w) + 1$, in particular, $d(y,v) \geq d(v,w)$.

If w and y are equidistant from v, then by the triangle condition there is a common neighbor z of w and y one step closer to v. Since $\alpha(y) < \alpha(x)$, by BFS we get $\alpha(z) < \alpha(v)$. Then we obtain a contradiction to the choice of the vertices

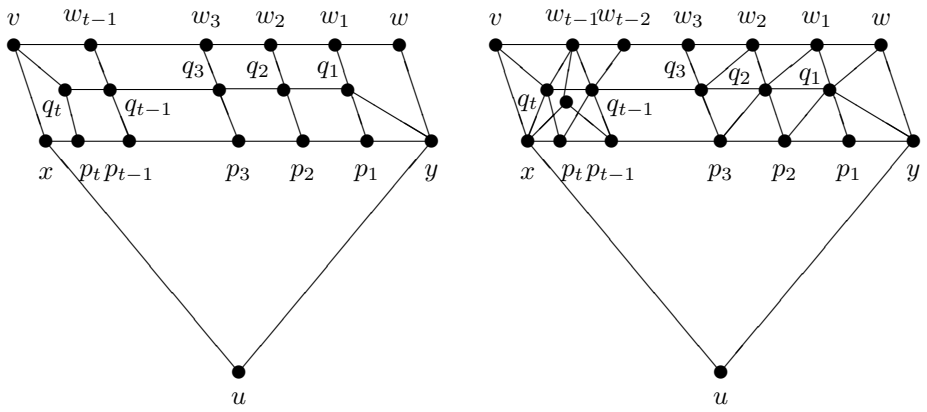


FIG. 4.1.

v and w, because $z \in I(v,w)$. Hence, $w \in I(y,v)$. Next, suppose that the vertex p also belongs to the interval $I(y,v)$. By the quadrangle condition there is a common neighbor $q \in I(w,v) \cap I(p,v)$ of w and p. And again, since $\alpha(p) < \max\{\alpha(y), \alpha(x)\} = \alpha(x)$, we conclude that $\alpha(q) < \alpha(v) = i$, contrary to our assumption. By Claim 2, $d(x,y) \le d(v,w) + 1$. If $d(x,y) = d(v,w)$, then $x \in I(v,y)$ and $p \in I(x,y) \subset I(v,y)$, which is impossible. Therefore, $d(x,y) = d(v,p) = d(v,w) + 1$, thus establishing Claim 2. □

We are now prepared to prove the theorem. Let $P_\alpha(y,x) = (y, p = p_1, p_2, \dots, p_t, x)$. Since $\alpha(p_j) < \alpha(x)$, from the choice of v and w and the procedure BFS we conclude that p_j cannot be adjacent to any vertex of the interval $I(v,w)$. As $d(v,y) = d(v,p_1)$, by the triangle condition there exists a common neighbor q_1 of p_1 and y which is one step closer to v. We assert that $q_1 \neq w$. Indeed, otherwise $p_2, w \in I(p_1,v)$ and by the quadrangle condition there is a common neighbor $w' \in I(w,v) \cap I(p_2,v)$ of w and p_2, contrary to our assumption. So, let $q_1 \neq w$. Then $w, q_1 \in I(y,v)$ and $q_1, p_2 \in I(p_1,v)$. Again applying the quadrangle condition, we can find the vertices $w_1 \in I(w,v) \cap I(q_1,v)$ and $q_2 \in I(q_1,v) \cap I(p_1,v)$, adjacent to w, q_1 and q_1, p_1, respectively; see Figure 4.1.

Recall that the vertices p_2 and w_1 cannot be adjacent. Therefore, $q_2 \neq w_1$. If $q_2 = p_3$, then $p_3, w_1 \in I(q_1,v)$. Applying the quadrangle condition we would get a common neighbor $w' \in I(w_1,v) \cap I(p_3,v)$ of w_1 and p_3. Since $\alpha(p_3) < \alpha(x)$, necessarily $\alpha(w') < \alpha(v)$, contrary to the fact that $w' \in I(v,w)$. So, let $q_2 \neq p_3$. Since $q_2, p_3 \in I(p_2,v)$ and $q_2, w_1 \in I(q_1,v)$, we could find the vertices q_3 and w_2 one step closer to v and adjacent to q_2, p_3 and q_2, w_1, respectively. Walking this way along the path $P_\alpha(x,y)$, we will find distinct vertices $w_3, \dots, w_{t-1}$ and $q_3, \dots, q_{t-1}, q_t$ such that $w, w_1, w_2, \dots, w_{t-1}, v$ induce a shortest path between w and v, and $y, q_1, q_2, \dots, q_{t-1}, q_t, v$ induce a shortest path between y and v. In addition, every q_j, $1 \le j < t$ is adjacent to w_j and p_j and $q_j \in I(p_j,v), w_j \in I(q_j,v)$, whereas the last vertex q_t is adjacent to p_t and v.

We claim that every q_j, $1 < j < t$, is adjacent to w_{j-1} and p_{j+1}, while q_1 is adjacent to w, p_2 and q_t is adjacent to x and w_{t-1}. Consider the subgraphs induced by the vertices w, w_1, q_1, y, p_1 and y, q_1, p_1, p_2, q_2. As we already established, w_1 and p_1 cannot be adjacent. Since G is house free, from the metric relations between the involved vertices and the vertex v we deduce that q_1 must be adjacent to p_2 and w. Again, in order to avoid induced houses, q_2 must be adjacent to w_1 and p_3.

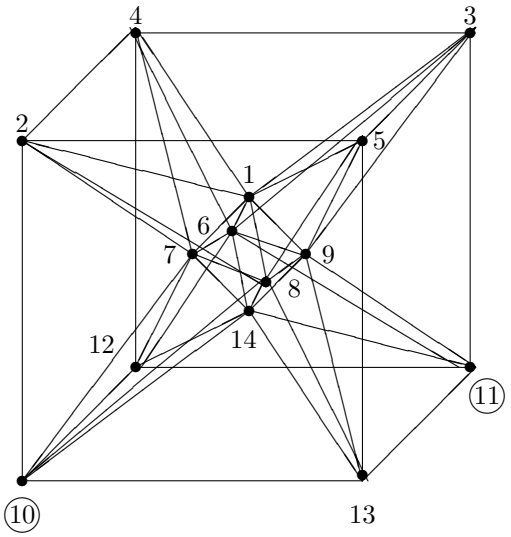


FIG. 4.2.

Continuing this way, we eventually obtain that q_{t-1} would be adjacent to p_t and w_{t-2}, while q_t would be adjacent to x and w_{t-1}; see Figure 4.1. The vertices v and p_{t-1} are at distance three. Since $x, w_{t-1} \in I(v, p_{t-1})$, we will find a common neighbor z of the vertices x, w_{t-1} and p_{t-1}. We assert that $z \neq q_{t-1}$. Indeed, if x and q_{t-1} were adjacent, we would get a path $y, q_1, q_2, \ldots, q_{t-1}, x$ of length $t = d(v, w)$ between x and y, contrary to Claim 2. Consider the subgraph induced by the vertices $w_{t-1}, w_{t-2}, z, q_{t-1}$, and p_{t-1}. Since the graph G is house free and the vertex p_{t-1} cannot be adjacent to w_{t-2} or w_{t-1}, we deduce that z is adjacent either to w_{t-2} or to q_{t-1}. In both cases we will get that x and w_{t-1} must be adjacent, otherwise we obtain a forbidden house induced by the vertices $v, x, z, w_{t-1}, w_{t-2}$ or $v, x, z, w_{t-1}, q_{t-1}$. Then, however, the vertices x, w_{t-1}, q_{t-1}, p_t and p_{t-1} induce a house. This final contradiction shows that the function α is pseudopeakless. By Lemma 3.2 the ordering produced by BFS is distance preserving. □

We do not know if the assertion of Theorem 4.2 is true for all weakly modular graphs. However, it is not true for meshed graphs. We take the 3-cube Q_3 and its dual graph (the 3-octahedron O_3) and join any vertex of O_3 to all vertices of Q_3 which belong to the respective face of Q_3; see Figure 4.2. A straightforward analysis shows that the resulting graph H is meshed. In Figure 4.2 we present a BFS ordering of H which is not distance preserving. Namely, if we consider the vertices with numbers 10 and 11, then all their common neighbors get a larger number.

COROLLARY 4.3. *For a graph G the following conditions are equivalent:*

(i) *for each isometric subgraph of G, the ordering of its vertices produced by BFS is distance preserving;*

(ii) *G is a hereditary weakly modular graph.*

Proof. Any cycle C_n, $n \geq 5$ does not have a distance-preserving ordering. Figure 5.2 provides a BFS ordering of the vertices of a house which is not distance preserving. Thus (i) → (ii). The converse follows from Theorem 4.2. □

5. Domination elimination orderings of hereditary weakly modular graphs. Next we will show that in hereditary weakly modular graphs and their subclasses the

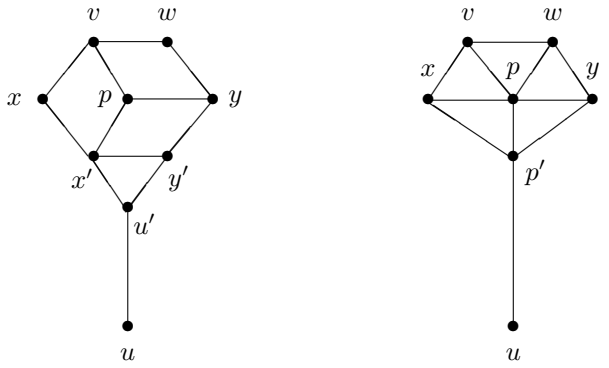


FIG. 5.1.

BFS and LBFS orderings have additional properties. First we need a lemma.

LEMMA 5.1. *Let v and w be two adjacent vertices of a hereditary weakly modular graph G which are equidistant to u. If x and y are the fathers of v and w, then either x and y are adjacent or coincide. In addition, if $\alpha(v) < \alpha(w)$, then y is adjacent to v.*

Proof. We proceed by induction on $k = d(u, v)$. If $d(u, v) = 1$, then $x = u = y$ and we are done. So, let $k > 1$. Suppose by way of contradiction that $d(x, y) > 1$. Since $\alpha(v) < \alpha(w)$, according to BFS $\alpha(x) < \alpha(y)$. Therefore, the vertices x and w must be nonadjacent. We distinguish two cases.

Case 1. $d(x, y) = 3$.

Then any pseudomedian of the triple u, x, and y has either size 3 or size 1. In either case such a pseudomedian consists of x, y and a vertex u' at distance 3 from x and y. From the properties of weakly modular graphs presented in section 2 we deduce that $d(v, u') = d(w, u') = 3$. Since $d(u', u) = k - 4$, we get a contradiction with $d(u, v) = d(u, w) = k$. So, assume that the pseudomedian x', y', and u' of the vertices x, y, and u has size 1; see Figure 5.1. Then $d(x', u) = d(y', u) = k - 2$ and $d(u', u) = k - 3$. Since $v, x' \in I(x, y)$, by the quadrangle condition there is a common neighbor p of v, x' and y. As G is house free, the vertices p and y' must be adjacent. But then the vertices x, v, x', p, and y' induce a house, because $d(x, y) = 3$ and $d(x', u) = d(y', u) = k - 3$.

Case 2. $d(x, y) = 2$.

Applying the triangle condition to the vertices x, y, w, we can find their common neighbor p. In order to avoid an induced house, the vertices p and v must be adjacent, too; see Figure 5.1. According to BFS $\alpha(p) > \alpha(y)$. We assert that $d(p, u) = k - 1$. Indeed, otherwise $d(p, u) = k$ and $x, y \in I(p, y)$. By the quadrangle condition there is a common neighbor q of x and y at distance $k - 2$ from u. Then we get two houses, induced by the vertices v, w, x, y, p, and q. So, let $d(p, u) = k - 1$. Consider the fathers x', p', and y' of the vertices x, p, and y, respectively. By the induction assumption, $d(x', p') \leq 1$ and $d(p', y') \leq 1$. In addition, p' must be adjacent to both x and y, because $\alpha(x) < \alpha(p) > \alpha(y)$. Then, however, the vertices p', x, v, w, y induce a 5-cycle, which is impossible. So, $d(x, y) \leq 1$.

Now, assume that y and v were nonadjacent. Since x and y are at distance $k - 1$ to u, by the triangle condition there is a common neighbor u' of x and y at distance $k - 2$ to u. As a result we will get a house, induced by v, w, x, y, and u'. □

COROLLARY 5.2 (see [1, 18]). *Bridged graphs are cop-win graphs. Moreover, any ordering of a bridged graph G produced by BFS is a cop-win ordering.*

Proof. As in [18], by induction on i we will show that the vertex $w = v_i$ is dominated in G_i by its father y. Pick an arbitrary neighbor v of w in G_i. If $d(v,u) = d(w,u)$, then y and v would be adjacent in view of Lemma 5.1. Otherwise, $y, v \in I(w,u)$, and, by the quadrangle condition there is a vertex u' adjacent to y and v and at distance $d(w,u) - 2$ to u. Since G does not contain induced 4-cycles, the vertices y and v must be adjacent. □

LEMMA 5.3. *If Γ is an induced 6-cycle of a hereditary weakly modular graph G, there is a vertex adjacent to all vertices of Γ. In particular, $d(u,v) \leq 2$ for any $u, v \in \Gamma$.*

Proof. As Γ cannot be isometric, necessarily two opposite vertices of Γ, say u and v, are at distance two. Let w be a neighbor of v in Γ. By weak modularity there is a common neighbor x of u, v, and w. In order to avoid the forbidden induced house or 5-cycle, x must be adjacent to the remaining vertices of Γ. □

For a vertex v with $\alpha(v) = i$ denote

$$N'(v) = \{w \in N_i(v) : d(w,u) < d(v,u)\},$$

$$N''(v) = N_i(v) - N'(v).$$

LEMMA 5.4. *Let v be a vertex of a hereditary weakly modular graph G at distance k from u. Then all neighbors of v in $I(v,u)$ are adjacent to some vertex v^* at distance $k-2$ to u.*

Proof. Let v^* be a vertex of $S_{k-2}(u)$ which is adjacent to a maximum number of vertices of $N'(v)$. Assume by way of contradiction that there is a neighbor w of v in $I(v,u)$ that is not adjacent to v^*. Pick an arbitrary vertex $z \in N(v) \cap N(v^*)$. As $z, w \in I(v,u)$, by the quadrangle condition there is a common neighbor $y \neq v^*$ of z and w at distance $k-2$ to u. And again, applying the quadrangle condition to the vertices $v^*, y \in I(z,u)$ we can find a common neighbor u' of v^* and y which is one step closer to u. From the choice of the vertex v^* we conclude that there exists a vertex $q \in N(v^*) \cap N(v)$ which is not adjacent to y. As a result we will get an induced 6-cycle (v, q, v^*, u', y, w). By Lemma 5.3 $d(v,u') = 2$, which is a contradiction. □

COROLLARY 5.5. *Any ordering of the vertices of a hereditary modular graph G produced by BFS is a domination elimination ordering.*

Proof. Since G is bipartite, for any vertex v with $\alpha(v) = i$ we have $N_i(v) = N'(v)$. By Lemma 5.4 we are done. □

Therefore, the simplest instances of hereditary weakly modular graphs have domination elimination orderings which can be computed by BFS. As the following example shows, this is not more true for all hereditary weakly modular graphs. Consider a graph $G = L(G_1, G_2)$ which consists of copies of the graphs G_1 and G_2 and all edges xy, where $x \in G_1$ and $y \in G_2$. We shall say that G is a *join* of the graphs G_1 and G_2. Notice that every such G is hereditary weakly modular, provided G_1 and G_2 do not contain 5-cycles and houses as induced subgraphs. Moreover, if both G_1 and G_2 are HC-free graphs, then G is also HC-free. The graph $L(C_n, C_m)$, $n \geq 6, m \geq 6$ is hereditary weakly modular; however, it does not have a domination elimination ordering. We can dismantle all hereditary weakly modular graphs by relaxing the domination condition. Namely, we shall say that an edge xy *dominates* a vertex w whenever $N[w] \subseteq N[x] \cup N[y]$. An ordering $v_1, v_2, \ldots, v_n$ of the vertices of G is called an *edge-dominating elimination ordering* if for every $v_i, i < n$, there exists an edge $v_j v_k, j < i, k < i$, which dominates the vertex v_i in G_i.

THEOREM 5.6. *For a house-free graph G the following conditions are equivalent:*

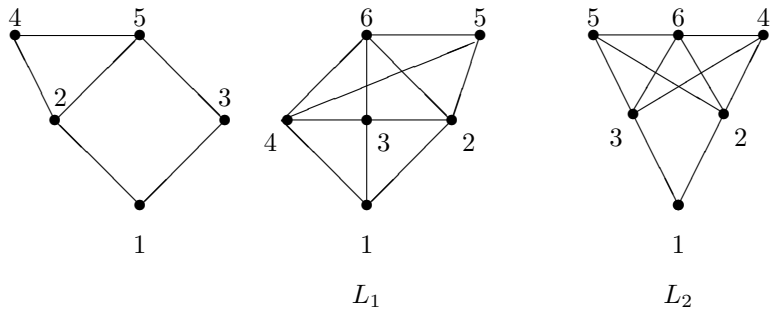


FIG. 5.2.

(i) *G and each isometric subgraph admit an edge-domination elimination ordering;*
(ii) *G is a hereditary weakly modular graph;*
(iii) *for each isometric subgraph of G, the ordering of its vertices produced by BFS is an edge-dominating ordering.*

Proof. (i) $\rightarrow$ (ii) and (iii) $\rightarrow$ (i) are evident.

(ii) $\rightarrow$ (iii). The hereditary property of G infers that it is sufficient to prove the result only for the whole graph G. In the ordering produced by BFS consider the current vertex v, where $\alpha(v) = i$. If $N_i(v) = N'(v)$, according to Lemma 5.4 v is dominated by any vertex v^*. Then v^* together with any $p \in N_i(v)$ forms an edge which dominates the vertex v. So, suppose that the set $N''(v)$ is nonempty. By Lemma 5.1 the father x of v will be adjacent to any vertex $w \in N''(v)$. We assert that the edge xw dominates the vertex v. Suppose the contrary: then there exists a vertex $z \in N'(v)$ nonadjacent to both x and w. Let y be a common neighbor of z and x one step closer to u. Then the vertices v, w, x, z, y induce a house, which is a contradiction. □

Next, we are interested in graphs with the property that each isometric subgraph has a domination elimination ordering and this ordering can be computed by BFS or LBFS.

THEOREM 5.7. *For a graph G the following conditions are equivalent:*
(i) *for each isometric subgraph of G, the ordering of its vertices produced by BFS is a domination elimination ordering;*
(ii) *G does not contain the house, the graphs L_1, L_2 (Figure 5.2), and any cycle $C_n, n \geq 5$, as an isometric subgraph;*
(iii) *G is a hereditary weakly modular graph which does not contain the graphs L_1 and L_2 as induced subgraphs.*

Proof. (i) $\rightarrow$ (ii) follows from Corollary 4.3, while (ii) $\rightarrow$ (iii) follows from the characterization of hereditary weakly modular graphs presented in section 2.

(iii) $\rightarrow$ (i). As in the preceding theorem, it suffices to establish the result only for the whole graph G. Consider a vertex v with $\alpha(v) = i$. Let $d(v, u) = k$. Again, if $N_i(v) = N'(v)$, by Lemma 5.4 v is dominated in G_i by some vertex v^* at distance $k - 2$ from u. So, we can suppose that the set $N''(v) = N_i(v) - N'(v)$ is nonempty. According to the proof of Theorem 5.6, any edge xw, where $w \in N''(v)$ and x is the father of v in the tree T_u, dominates the vertex v. Assume that separately the vertices x and w do not dominate the vertex v. Therefore, we can find two vertices $p, q \in N_i(v)$ such that p and x on the one hand, and q and w on the other hand, are nonadjacent.

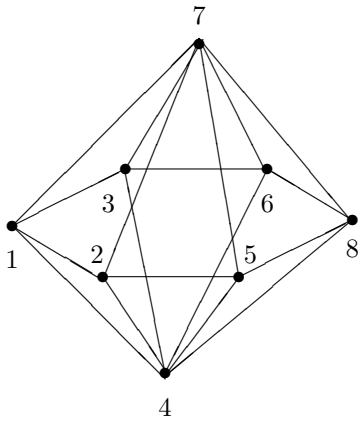


FIG. 5.3.

We assert that $p \neq q$. Indeed, otherwise the vertices v, w, x, v^*, p induce a house. As x is adjacent to all vertices of $N''(v)$, necessarily $p \in I(v, u)$. So, $p \neq q$, $p \in I(v, u)$, and the vertices p and w are adjacent. If $q \in N'(v)$, the vertex q must be adjacent to both x and p, otherwise we get a house induced by the vertices v, q, x, v^*, w or by the vertices v, x, p, v^*, w. However, in this case the vertices v, w, x, p, q, v^* induce a forbidden subgraph L_1. Now, let $q \in N''(v)$. Again, the vertices p and q must be adjacent, otherwise q, v, x, p, and v^* induce a house. In this case we will get the second forbidden subgraph L_2. □

Dahlhaus et al. [21] introduced the concept of a *domination graph* as a graph with the property that each induced subgraph has a domination elimination ordering. As follows from [21], HC-free graphs are exactly the house-free domination graphs. Moreover, a domination elimination ordering of HC-graphs can be computed by MCS or LBFS [21]. Further, we characterize house-free graphs with the property that any LBFS ordering of each isometric subgraph is a domination elimination ordering. In this case a graph can contain induced cycles of any length $\neq 5$. However, two restrictions on induced cycles are imposed. First, any induced cycle $C_n, n \geq 6$ cannot be isometric. Second, it cannot occur as a cycle in a bipyramid. A *bipyramid* $bipyr(C_m)$ is the graph $L(C_m, \overline{K_2})$, i.e., it consists of a cycle of length m and two nonadjacent vertices which are adjacent to all vertices of this cycle. Figure 5.3 presents an LBFS ordering of the bipyramid $bipyr(C_6)$, which is not a domination elimination ordering.

THEOREM 5.8. *Let G be a house-free graph. The following conditions are equivalent:*

(i) *for each isometric subgraph of G, the ordering $v_k, v_{k-1}, \ldots, v_1$ of its vertices produced by LBFS is a domination elimination ordering;*
(ii) *G does not contain any cycle $C_m, m \geq 5$, and any bipyramid $bipyr(C_m), m \geq 6$ as an isometric subgraph;*
(iii) *G is a hereditary weakly modular graph which does not contain any bipyramid $bipyr(C_m), m \geq 6$, as an induced subgraph.*

Proof. It suffices to establish that (iii) → (i), namely, that any LBFS ordering of a hereditary weakly modular graph G without bipyramids $bipyr(C_m), m \geq 6$ as an induced subgraph is a domination elimination ordering. Let the notations remain as before, only $N_i(v_j) = N(v_j) \cap \{v_i, v_{i+1}, \ldots, v_n\}$ and $G_i = G(v_i, \ldots, v_n)$. Again, we suppose that the procedure LBFS has the vertex u as the starting point, i.e., $\alpha(u) = n$.

We proceed by induction on the number n of vertices of G. By Theorem 4.2 every G_i is an isometric subgraph of G. Therefore, it suffices to prove that the vertex v with $\alpha(v) = 1$ is dominated by some vertex of G. Let $d(v,u) = k$. We can suppose that $k \geq 2$, otherwise v is dominated by u, and we are done. Throughout the proof for a vertex p by p^* we will denote a vertex at distance $d(p,u) - 2$ to u which is adjacent to all vertices of $N'(p)$ (see Lemma 5.4).

Assume by way of contradiction that no vertex of G dominates the vertex v. Then Lemma 5.4 implies that the set $N''(v) = N(v) - N'(v)$ is nonempty. By Lemma 5.1 the father f of v is adjacent to all vertices of $N''(v)$. Therefore, we could find a vertex $f^+ \in N'(v)$ which is nonadjacent to f. Then any $w \in N''(v)$ must be adjacent to f^+, otherwise the vertices w, v, f, f^+, v^* induce a house. Our proof requires a number of auxiliary results and notations that we present next.

Claim 1. *Let $w, z \in S_k(u)$ be two adjacent vertices with $\alpha(z) > \alpha(w)$, and let $N = N'(w) \cap N'(z)$. If N contains two nonadjacent vertices, then $N'(w) \subseteq N'(z)$ and any vertex of $N'(z) - N'(w)$ is adjacent to all vertices of $N'(w)$. Otherwise, if N is a clique, then any vertex of N is adjacent to all vertices of $N'(w) \cup N'(z)$.*

Proof of Claim 1. Since $d(w,u) = d(z,u) = k$, by the triangle condition there is a common neighbor of w and z at distance $k-1$ to u, i.e., the set N is nonempty. If N is a clique, but two vertices $t \in N$ and $s \in N'(w) - N'(z)$ are nonadjacent, then z, w, t, s, and w^* induce a house. So, suppose that N contains two nonadjacent vertices x and y; however, the sets $N'(w)$ and $N'(z)$ are incomparable. Then we can find vertices $p \in N'(w) - N'(z)$ and $q \in N'(z) - N'(w)$. By Lemma 5.4 there exists vertices w^* and z^* at distance $k-2$ to u, which are adjacent to the vertices p, x, y and q, x, y, respectively; see Figure 5.4. Necessarily, p and q must be adjacent to both x and y, otherwise we would get an induced house. First suppose that $w^* \neq z^*$, i.e., the vertices w^*, q and z^*, p are nonadjacent. As $w^*, z^* \in I(x,u)$, by the quadrangle condition there is a common neighbor $u^+ \in I(w^*,u) \cap I(z^*,u)$ of w^* and z^*. In order to avoid an induced house, the vertices w^* and z^* must be adjacent. If p and q were nonadjacent, we will get a bipyramid $bipyr(C_6)$ induced by p, q and the 6-cycle $(w, z, y, z^*, w^*, x, w)$. Otherwise, if p and q are adjacent, the vertices p, q, z^*, w^*, and u^+ induce a house. So, let $w^* = z^*$. However, then we get either an induced 5-cycle, or, if p and q are adjacent, an induced house. The obtained contradiction implies that $N'(w)$ and $N'(z)$ must be comparable. Since $\alpha(z) > \alpha(w)$, by LBFS we deduce that $N'(w) \subseteq N'(z)$. Now, pick two arbitrary vertices $s \in N'(z) - N'(w)$ and $t \in N'(w)$. If s and t were nonadjacent, then the vertices z, w, s, t, and z^+ induce a house, which is impossible. □

According to Claim 1 and LBFS, $N'(v) \subseteq N'(w)$ for any $w \in N''(v)$ (recall that w is adjacent to nonadjacent vertices f and f^+ of $N'(v)$). Hence, for any vertex $p \in N'(v)$ there is a vertex $p^+ \in N'(v)$ that is not adjacent to p. Denote by S the vertices $t \notin N[v]$ of G which are adjacent to all vertices of $N'(v)$. We let T be the union of S and $N[v]$.

Claim 2. *Let w and z be two adjacent vertices of $S_k(u)$. If $\alpha(z) > \alpha(w)$ and $N'(w) = N'(v)$, then $z \in T$.*

Proof of Claim 2. By the triangle condition there is a common neighbor p of w and z at distance $k-1$ to u. Since $p \in N'(v)$, we can find a vertex $p^+ \in N'(v)$ that is not adjacent to p. Then z and p^+ must be adjacent, otherwise we will obtain an induced house. By Claim 1 and because $\alpha(z) > \alpha(w)$, we conclude that $N'(v) = N'(w) \subseteq N'(z)$. Thus $z \in T$. □

Claim 3. *Every vertex $t \in N''(v)$ is adjacent to some vertex z of S with $\alpha(z) > \alpha(t)$.*

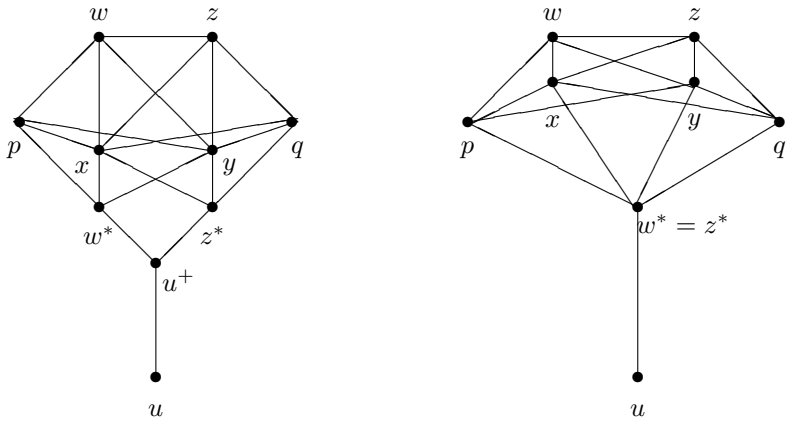


FIG. 5.4.

Proof of Claim 3. Since t does not dominate the vertex v, there is a vertex $t^+ \in N''(v)$ that is not adjacent to t. First, suppose that $\alpha(t^+) > \alpha(t)$. Then according to LBFS in step $\alpha(t)$ the label of t must be larger than that of v. Therefore, we can find a vertex $z \in N(t) - N(v)$ with $\alpha(z) > \alpha(t)$. We assert that $z \in S$. If $z \in N'(t)$, then z would be adjacent to any vertex $p \in N'(v) \subseteq N'(t)$, otherwise t, v, z, p, and t^* induce a house. So, let $z \in N''(t)$ and $N'(t) = N'(v)$. By Claim 2 we conclude that $z \in S$.

Now, assume that $\alpha(t) > \alpha(t^+)$. According to LBFS in step $\alpha(t)$ the labels of the vertices t and v either coincide or t has a larger label. In the second case t will be adjacent to a vertex $z \in N(t) - N(v)$ with $\alpha(z) > \alpha(t)$. By Claims 1 and 2 z has the required property. So, consider either case. According to LBFS in step $\alpha(t)$ the labels of all vertices q with $\alpha(t) > \alpha(q) \geq \alpha(v)$ coincide with that of vertex t. Moreover, all such q must be adjacent to t, because v and t are adjacent. Then we get a contradiction, since $\alpha(t) > \alpha(t^+) > \alpha(v)$, but t and t^+ are not adjacent. □

CLAIM 4. *The subgraph $G(T)$ induced by T is an HC-free graph.*

Proof of Claim 4. First, note that the subgraphs induced by $N'(v)$ and $T - N'(v)$ are HC-free. Indeed, otherwise we will get a forbidden bipyramid induced by either v, v^* and a cycle of length ≥ 6 of $G(N'(v))$ or by a similar cycle of $G(T - N'(v))$ and the vertices f and f^+. Now we are done, because $G(T)$ is the join of the graphs $G(N'(v))$ and $G(T - N'(v))$. □

From Claim 4 and Lemma 6 of [21] we obtain the following property of the subgraph $G(T)$.

CLAIM 5. *For each connected component C of $G(S) = G(T - N[v])$ there is a vertex p_C in C, such that*

$$N(C) \cap N(v) = N(p_C) \cap N(v).$$

Let w be the vertex with the greatest index $i = \alpha(w)$ over all $\alpha(p), p \in T_0$, where T_0 is the connected component of $T \cap S_k(u)$ containing the vertex v.

CLAIM 6. *The vertex w is adjacent to a vertex $q \in S \cap (N'(w) - N'(v))$.*

Proof of Claim 6. First we will show that $N'(w) - N'(v) \neq \emptyset$. Suppose the contrary, i.e., $N'(w) = N'(v)$. Then in step i of LBFS the labels of the vertices w and v must be equal to $N'(v)$. Indeed, otherwise we could find a vertex $z \in N(w) - N(v)$ with $\alpha(z) > i$. Since $d(z, u) = k$, by Claim 2 we obtain that $z \in S$. Consequently,

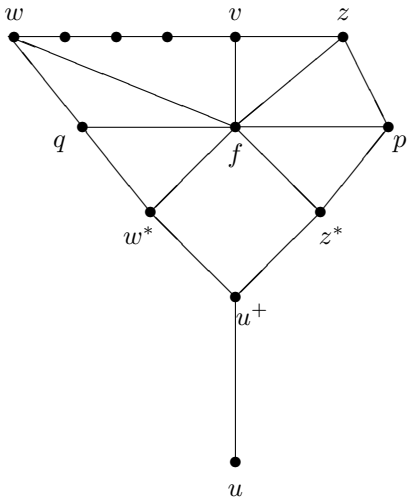


FIG. 5.5.

$z \in T_0$, contrary to the choice of the vertex w. Hence, in step i the set $N'(v)$ represents the label of each vertex of T_0. It may be observed that the restriction of the function α on the set T_0 can be derived by applying LBFS to the graph $G(T_0)$. More exactly, there is a numbering of the vertices of T_0 produced by LBFS that is equivalent to $\alpha|_{T_0}$. The graph $G(T_0)$ is hereditary weakly modular, because it is an induced subgraph of the HC-free graph $G(T)$. By the induction assumption, the vertex v is dominated in $G(T_0)$ by some vertex. However, then the same vertex still dominates v in the whole graph G, contrary to our initial assumption.

Pick an arbitrary vertex $q \in N'(w) - N'(v)$. We assert that $q \in S$. Let P be an induced path connecting the vertices w and v inside T_0. Recall that every vertex of P is adjacent to all vertices of $N'(v)$. By Claim 1 we could find two adjacent vertices t and s of P such that $q \in N'(t) - N'(s)$. Again, Claim 1 implies that q must be adjacent to all vertices of $N'(s) \supseteq N'(v)$, i.e., $q \in S$. □

Let the notations remain as before, and consider the connected component C^* of $G(S)$ containing the vertex q defined in Claim 6. Since by Claim 5 all vertices of $N(v) \cap N(C)$ are adjacent to a vertex $p_{C^*} \in C$, our initial assumption infers that $N(v) - N(C^*) \neq \emptyset$, otherwise p_{C^*} will dominate the vertex v. By Claim 3 any vertex $t \in N(v) - N(C^*)$ has a neighbor in some connected component C_t of the graph $G(S)$.

First suppose that there still exists a component C_t that contains a vertex p at distance $k-1$ to u. Evidently, p can be selected to be adjacent to a vertex $z \in T_0$. Namely, either z is a vertex of C_t, or, if $C_t \cap T_0 = \emptyset$, we can let $z = t$. Consider the vertices w^* and z^*. They must be distinct and nonadjacent, because $w^* \in C^*$ and $z^* \in C_t$. Since $w^*, z^* \in I(f,u)$, there is a vertex $u^+ \in I(w^*,u) \cap I(z^*,u)$ that is adjacent to w^* and z^*. Then, however, we get two houses induced by the vertices q, f, w^*, z^*, u^+, and p; see Figure 5.5.

Thus we can assume that each connected component $C_t, t \in N(v) - N(C^*)$, is entirely contained in the sphere $S_k(u)$. In particular, $N'(z) = N'(v)$ for any $z \in \cup\{C_t : t \in N(v) - N(C)\}$. Among such vertices z take the vertex z^+ with the greatest index $j = \alpha(z^+)$. We assert that in step j of LBFS the vertices z^+ and v have equal labels. Otherwise, there is a vertex $s \in N(z^+) - N(v)$ with $\alpha(s) > j$. As $s \in S_k(u)$, by Claim

2 we conclude that $s \in S$. Since $s \notin N[v]$, necessarily s and z^+ belong to a common connected component C_t, contrary to the choice of the vertex z^+.

Hence, in step j all vertices $p \in T_0$ with $\alpha(p) \leq j$ have one and the same label. It consists of $N'(v)$ and all neighbors r of v in $S_k(u)$ with the property that $\alpha(r) > j$. Denote the collection of such vertices r by R. Assume that $R = \emptyset$. Since the connected component C^* contains the vertex $q \in S_{k-1}(u) - N'(u)$ with $\alpha(q) > j$, we conclude that LBFS numbers all vertices of C^* before z^+. Therefore, in step j the label of any vertex from $N(C^*) \cap N(v)$ must be larger than that of z^+, i.e., any such vertex must be already numbered. This implies that R is nonempty. Moreover, since v is not dominated, for any $r \in R$, there exists a vertex $r^+ \in R$ that is not adjacent to r. On the other hand, at least one vertex of $N''(v)$ must be outside R, otherwise z^+ dominates v.

Finally, consider a new graph G' obtained from the subgraph of G induced by $R \cup \{p \in T_0 : \alpha(p) \leq j\}$ by adding a new vertex y that has R as its neighborhood and by removing all edges between vertices of R. Evidently, G' is a HC-free graph, because it is the join of an edgeless graph R and an HC-graph $G(\{p \in T_0 : \alpha(p) \leq j\}) + y$. It may be observed that the ordering α of the vertices of the set $R \cup \{p \in T_0 : \alpha(p) \leq j\}$ may be obtained as a LBFS ordering of the vertices of G'. Indeed, LBFS numbers the vertex y first—after that the vertices of R in accordance with α (recall that in G' all vertices of R are pairwise nonadjacent). Finally, it numbers the vertices from $\{p \in T_0 : \alpha(p) \leq j\}$ exactly repeating the corresponding steps of the procedure LBFS in G. Since G' has less than n vertices, by the induction hypothesis the vertex v is dominated in G' by some vertex t. Evidently, $t \neq y$, because $N''(v) - R \neq \emptyset$. Since $G' - y$ is a partial subgraph of G and $N(v)$ is completely included in G', we deduce that t dominates the vertex v in the whole graph G. With this the proof of the theorem is complete. □

6. Lexicographic orderings of hypercubes. A *d-dimensional hypercube* Q_d (*d-cube*) is a graph whose vertices are all (0,1)vectors of length d, two vertices being joined if they differ in exactly one coordinate. Clearly Q_d is a bipartite d-regular graph with $n = 2^d$ vertices and $m = d \cdot 2^{d-1}$ edges. In the following we assume that only the adjacency matrix of Q_d is available.

The usual shortest-path distance between any two vertices u and v of Q_d is the number of positions in which u and v differ (this distance is called the *Hamming distance* between u and v). Any subcube of Q_d is called a *face* of Q_d. Any hypercube is a median graph [30].

Let T be a rooted tree. Recall that the *height* of T is the largest distance from the root to a leaf of T. The *height* of a vertex v is the height of the subtree of T with root v. Then the leaves of T are exactly the vertices of height 0, while the root of T is the unique vertex of maximum height.

THEOREM 6.1. *Let T be the spanning tree of the d-cube Q_d constructed by LBFS. Then the leaves of T induce in Q_d a $(d-1)$-cube. More generally, the vertices of height h of T induce a $(d-h-1)$-dimensional face of Q_d.*

Proof. Let u be the root of the tree T, i.e., $\alpha(u) = n$. Denote by L and $I = T - L$ the leaves and the interior vertices of the tree T. First we will show that for any vertex $v \in I$ its neighbor v' in Q_d with the smallest number in the LBFS ordering is a leaf of T and v is the father of v'. Suppose the contrary, and let v' be the father of some other vertex w. Consider the second common neighbor v'' in Q_d of v and w. Since $\alpha(v') < \alpha(v'')$, we obtain a contradiction with the fact that v' is the father of w.

Therefore, v' must be a leaf of T. In addition, since v is the father of some vertex

z and $\alpha(v') < \alpha(z)$, by LBFS we conclude that v is also the father of v'. We will say that the vertices v and v' form a *peripheral pair*.

Consider the peripheral pair (u, u'). We can split Q into two disjoint $(d-1)$-cubes Q and Q' with respect to the unique coordinate where u and u' differ. Let us assume that $u \in Q$ and $u' \in Q'$. Taking all edges with one end in Q and another one in Q' we will get a perfect matching of Q_d. We assert that every peripheral pair (v, v') defines an edge in this matching with $v \in Q$ and $v' \in Q'$. Pick an arbitrary vertex $v \in I$. We apply an induction argument on the distance $d(u, v)$ departing from $d(u, v) = 0$. The vertex v necessarily belongs to Q. Indeed, the father w of v being closer to u lies in Q, while the leaf w' is a vertex of Q'. Since $v \neq w'$ and w' is a unique neighbor of w in Q', we deduce that $v \in Q$. Suppose by way of contradiction that the vertex v' also belongs to Q. Let x be the neighbor of v in Q'. According to LBFS, in the step where we number x, the label of this vertex must be larger or equal to the labels of vertices numbered later. At this moment $\mathsf{label}(x)$ consists of v and all neighbors of x on shortest paths between x and u'. On the other hand, the label $\mathsf{label}(v')$ of v' comprises all neighbors of v' at distance $d(u, v)$ to u. Evidently, $\mathsf{label}(x) \subset Q' \cup \{v\}$, while $\mathsf{label}(v') \subset Q$. Pick an arbitrary vertex $y \in \mathsf{label}(x)$, $y \neq v$. Let z be the neighbor of y in Q. Notice that z cannot be a leaf of T, otherwise the peripheral pair consisting of z and its father would violate the induction assumption. Hence, $z \in I$, whence by induction assumption (z, y) is a peripheral pair. Let t denote the second common neighbor of the vertices z and v' in Q_d. Evidently, $t \in \mathsf{label}(v')$, because $d(t, u) = d(v, u)$. Since y is the neighbor of z with the minimum number, necessarily $\alpha(y) < \alpha(t)$. Therefore, the label of v' is larger than that of y, contrary to the fact that y is numbered before v'. This shows that $v' \in Q'$. Every interior vertex is a member of at least one peripheral pair, whence $I \subseteq Q$ and $Q' \subseteq L$. Since the father of any vertex of Q' will be its unique neighbor in Q, we obtain that $I = Q$ and $L = Q'$, concluding the proof of the first assertion.

Next, observe that applying LBFS to the $(d-1)$-cube Q we can construct the same spanning tree of Q as the subtree of T induced by its interior vertices. It remains to notice that all vertices of height $h > 0$ of T in the new tree have height $h-1$. So, to establish the second assertion we can proceed by induction on the dimension d. □

In order to recognize whether a given connected graph G is a d-cube, first we construct by LBFS a spanning tree T of G. After that we arrange the vertices of G according to their heights in T. Necessarily, G must have 2^{d-h-1} vertices of height h. Starting with the root of T, for a current h we have to check if the edges of T between the vertices of height h and their fathers define a perfect matching and an isomorphism between the subgraph of G induced by the vertices of height h and the subgraph induced by the vertices of height larger than h. The complexity of the given step is proportional to the number of edges of the second subgraph (it contains exactly $(d-h-1) \cdot 2^{d-h-1}$ edges). Therefore, the total complexity is linear with respect to the size of G. The graph G is the d-cube if and only if G passes each test. For another linear test see [11]; some characterizations of hypercubes are given in [30].

In Figure 6.1 we present an ordering and a spanning tree of the 4-cube Q_4 produced by the LBFS procedure. The leaves of this tree generate a three-dimensional face of Q_4.

We conclude with a BFS ordering of the 3-cube Q_3 so that the leaves of the tree T do not induce a face of Q_3; see Figure 6.2.

Acknowledgments. I wish to express my appreciation to Hans-Jürgen Bandelt for valuable discussions on the theory of discrete metric spaces. Also I am indebted

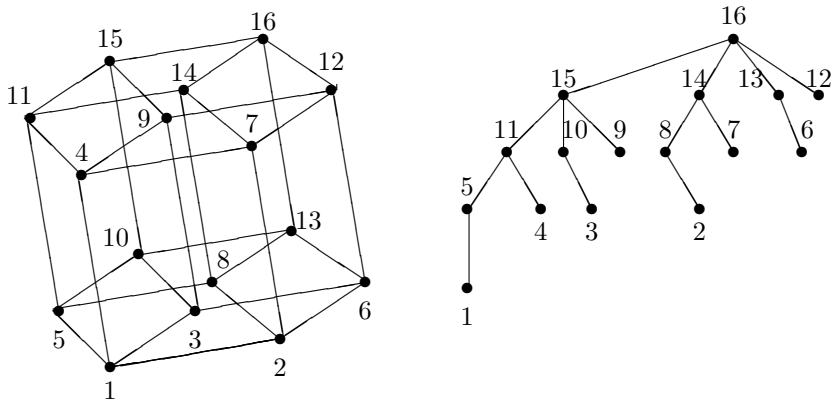


FIG. 6.1.

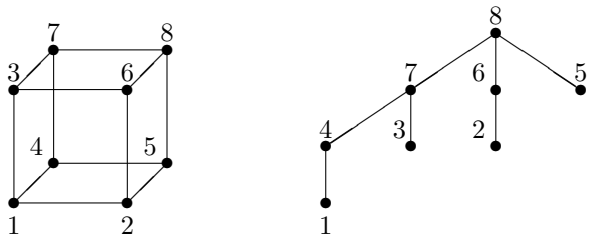


FIG. 6.2.

to Feodor Dragan for acquaintance with reference [21].

REFERENCES

[1] R. P. ANSTEE AND M. FARBER, *On bridged graphs and cop-win graphs*, J. Combin. Theory Ser. B, 44 (1988), pp. 22–28.
[2] H.-J. BANDELT, *Hereditary modular graphs*, Combinatorica, 8 (1988), pp. 149–157.
[3] H.-J. BANDELT AND V. CHEPOI, *A Helly theorem in weakly modular space*, Discrete Math., 126 (1996), pp. 25–39.
[4] H.-J. BANDELT AND J. HEDLÍKOVÁ, *Median algebras*, Discrete Math., 45 (1983), pp. 1–30.
[5] H.-J. BANDELT AND H. M. MULDER, *Distance-hereditary graphs*, J. Combin. Theory Ser. B, 41 (1986), pp. 182–208.
[6] H.-J. BANDELT AND H. M. MULDER, *Pseudo-modular graphs*, Discrete Math., 62 (1986), pp. 245–260.
[7] H.-J. BANDELT AND H. M. MULDER, *Pseudo-median graphs: Decomposition via amalgamation and Cartesian multiplication*, Discrete Math., 94 (1991), pp. 161–180.
[8] H.-J. BANDELT AND H. M. MULDER, *Cartesian factorization of interval-regular graphs having no long isometric odd cycles*, in Graph Theory, Combinatorics, and Applications, Vol. 1, Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, eds., John Wiley, New York, 1991, pp. 55–75.
[9] H.-J. BANDELT, H. M. MULDER, AND V. SOLTAN, *Weak Cartesian Factorization with Icosahedra, 5-wheels, and Subhyperoctahedra as Factors*, Econometric Institut report 9455, Erasmus University, Rotterdam, the Netherlands, 1994.
[10] H.-J. BANDELT AND E. PESCH, *Dismantling absolute retracts of reflexive graphs*, European J. Combin., 10 (1989), pp. 211–220.
[11] K. V. S. BHAT, *On the complexity of testing a graph for n-cube*, Inform. Process. Lett., 11 (1980), pp. 16–19.
[12] H. BUSEMANN, *The Geometry of Geodesics*, Academic Press, New York, 1955.

[13] H. Busemann and B. B. Phadke, *Peakless and monotone functions on G-spaces*, Tsukuba J. Math., 7 (1983), pp. 105–135.
[14] H. Busemann and B. B. Phadke, *Novel results in the geometry of geodesics*, Adv. Math., 101 (1993), pp. 180–219.
[15] J. W. Cannon, *Almost convex groups*, Geom. Dedicata, 22 (1987), pp. 197–210.
[16] V. Chepoi, *Classifying graphs by metric triangles*, Metody Diskret. Anal., 49 (1989), pp. 75–93 (in Russian).
[17] V. Chepoi, *Peakless functions on graphs*, Discrete Appl. Math., 73 (1997), pp. 175–189.
[18] V. Chepoi, *Bridged graphs are cop-win graphs: An algorithmic proof*, J. Combin. Theory Ser. B, 69 (1997), pp. 97–100.
[19] F. Dragan, F. Nicolai, and A. Brandstädt, *Convexity and HHD-free Graphs*, Tech. report SM-DU-290, Gerhard-Mercator-Universität-Gesamthochschule, Duisburg, Germany, 1995.
[20] F. Dragan, *Personal communication*, 1995.
[21] E. Dahlhaus, P. Hammer, F. Maffray, and S. Olariu, *On domination elimination orderings and domination graphs*, in Graph-Theoretic Concepts in Computer Science, WG'94, Lecture Notes in Comput. Sci., 903, Springer-Verlag, New York, 1994, pp. 81–92.
[22] M. Farber and R. E. Jamison, *Convexity in graphs and hypergraphs*, SIAM J. Algebraic Discrete Meth., 7 (1986), pp. 433–444.
[23] M. Farber and R. E. Jamison, *On local convexity in graphs*, Discrete Math., 66 (1987), pp. 231–247.
[24] B. Jamison and S. Olariu, *On the semi-perfect elimination*, Adv. Appl. Math., 9 (1988), pp. 364–376.
[25] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[26] P. Hammer, B. Simeone, T. Liebling, and D. de Werra, *From linear separability to unimodality: A hierarchy of pseudo-Boolean functions*, SIAM J. Discrete Math., 1 (1988), pp. 174–184.
[27] P. Hammer and F. Maffray, *Completely separable graphs*, Discrete Appl. Math., 27 (1990), pp. 85–99.
[28] R. B. Hayward, *Weakly triangulated graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 200–208.
[29] E. Howorka, *A characterization of distance-hereditary graphs*, Quart. J. Math. Oxford Ser. 2, 28 (1977), pp. 417–420.
[30] H. M. Mulder, *The Interval Function of a Graph*, Math. Centre Tracts 132, Amsterdam, 1980.
[31] R. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math., 43 (1983), pp. 235–239.
[32] R. Nowakowski and I. Rival, *The smallest graph variety containing all paths*, Discrete Math., 43 (1983), pp. 223–234.
[33] A. Quilliot, *On the Helly property working as a compactness criterion on graphs*, J. Combin. Theory Ser. A, 40 (1985), pp. 186–193.
[34] A. Quilliot, *Homomorphismes, points fixes, rétractions et jeux de poursuite dans les graphes, les ensembles ordonnés et les espaces métriques*, Thése d'Etat, Université de Paris VI, Paris, France, 1983.
[35] D. Rose, R. Tarjan, and G. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[36] R. Scharlau, *Metrical shellings of simplicial complexes*, European J. Combin., 6 (1985), pp. 265–269.
[37] D. R. Shier, *Some aspects of perfect elimination orderings in chordal graphs*, Discrete Appl. Math., 7 (1984), pp. 325–331.
[38] V. Soltan and V. Chepoi, *Conditions for invariance of set diameters under d-convexification in a graph*, Cybernetics, 19 (1983), pp. 750–756.
[39] R. E. Tarjan and M. Yannakakis, *Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.
[40] G. Ziegler, *Lectures on Polytopes*, Springer-Verlag, New York, 1994.
[41] K. H. Williamson, *Completely unimodal numberings of a simple polytope*, Discrete Appl. Math., 20 (1988), pp. 69–81.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 437–455, August 1998

007

DUALLY CHORDAL GRAPHS*

ANDREAS BRANDSTÄDT†, FEODOR DRAGAN†‡, VICTOR CHEPOI‡§, AND VITALY VOLOSHIN‡

Abstract. Recently in several papers, graphs with maximum neighborhood orderings were characterized and turned out to be algorithmically useful. This paper gives a unified framework for characterizations of those graphs in terms of neighborhood and clique hypergraphs which have the Helly property and whose line graph is chordal. These graphs are dual (in the sense of hypergraphs) to chordal graphs. By using the hypergraph approach in a systematical way new results are obtained, some of the old results are generalized, and some of the proofs are simplified.

Key words. graphs, hypergraphs, tree structure, hypertrees, duality, chordal graphs, clique hypergraphs, neighborhood hypergraphs, disk hypergraphs, Helly property, chordality of line graphs, maximum neighborhood orderings, linear time recognition, doubly chordal graphs, strongly chordal graphs, bipartite incidence graphs

AMS subject classifications. 05C65, 05C75, 68R10

PII. S0895480193253415

1. Introduction.

The class of chordal graphs is a by now classical and well-understood graph class which is algorithmically useful and has several interesting characterizations. In the theory of relational database schemes there are close relationships between desirable properties of database schemes, acyclicity of corresponding hypergraphs, and chordality of graphs which corresponds to tree and Helly properties of hypergraphs [2], [5], [25]. Chordal graphs arise also in solving large sparse systems of linear equations [28], [36] and in facility location theory [13].

Recently a new class of graphs was introduced and characterized in [20], [6], [21], [39] which is defined by the existence of a maximum neighborhood ordering. These graphs appeared first in [20] and [16] under the name *HT–graphs* but only a few results have been published in [21]. [34] also introduces maximum neighborhoods but only in connection with chordal graphs (chordal graphs with maximum neighborhood ordering were called there *doubly chordal graphs*).

It is our intention here to attempt to provide a unified framework for characterizations of those graph classes in terms of neighborhood and clique hypergraphs. These graphs are dual (in the sense of hypergraphs) to chordal graphs (this is why we call them *dually chordal*) but have very different properties—thus they are in general not perfect and not closed under taking induced subgraphs. By using the hypergraph approach in a systematical way new results are obtained, a part of the previous results are generalized, and some of the proofs are simplified. The present paper improves the results of the unpublished manuscripts [20] and [6].

Graphs with maximum neighborhood orderings (alias dually chordal graphs) are a generalization of strongly chordal graphs (a well-known subclass of chordal graphs

* Received by the editors August 6, 1993; accepted for publication (in revised form) August 19, 1997; published electronically July 7, 1998.
http://www.siam.org/journals/sidma/11-3/25341.html
† Universität Rostock, Fachbereich Informatik, Albert-Einstein-Str. 21, D-18051 Rostock, Germany (ab@informatik.uni-rostock.de)
‡ Department of Mathematics and Cybernetics, Moldova State University, A. Mateevici Str. 60, Chişinău 277009, Moldova (dragan@informatik.uni-rostock.de, chepoi@Mathematik.Uni-Bielefeld.DE, vol@usm.md)
§ Current address: Laboratoire de Biomathématiques, Université d'Aix Marseille II, 27 Bd Jean Moulin, F–13385 Marseille Cedex 5, France.

for which not only a maximum neighborhood but a linear ordering of neighborhoods of neighbors is required—this leads to the fact that strongly chordal graphs are exactly the hereditary dually chordal graphs, i.e., graphs for which each induced subgraph is a dually chordal graph). Notice also that doubly chordal graphs are precisely those graphs which are chordal and dually chordal.

Maximum neighborhood orderings are also algorithmically useful, especially for domination-like problems and problems which are based on distances. Many problems remaining NP–complete on chordal graphs have efficient algorithms on strongly chordal graphs. In some cases this is due to the existence of maximum neighbors (and not to chordality). Therefore many problems efficiently solvable for strongly chordal and doubly chordal graphs remain polynomial-time solvable for dually chordal graphs, too. In the companion papers [18], [19], [9], [10] the algorithmic use of the maximum neighborhood orderings is treated systematically. Dually chordal graphs seem to represent an important supplement of the world of classical graph classes.

One of our theorems shows that a graph G has a maximum neighborhood ordering if and only if the neighborhood hypergraph of G is a *hypertree*, i.e., it has the Helly property and its line graph is chordal. Due to the self-duality of neighborhood hypergraphs this is also equivalent to the α–acyclicity of the hypergraph which implies a linear time recognition of the graph class. This contrasts with the fact that the best known recognition algorithms for strongly chordal graphs have complexity $O(|E|log|V|)$ [35] and $O(|V|^2)$ [38].

There are several interesting generalizations of this class. Theorem 4 shows that a graph G has a maximum neighborhood ordering if and only if the clique hypergraph (or the disk hypergraph) of G has the Helly property and its line graph is chordal. It is known from [4], [17] that G is a disk–Helly graph (i.e., a graph whose disk hypergraph has the Helly property) if and only if G is a dismantlable clique–Helly graph, and in [3] it is shown that G is an absolute reflexive retract if and only if G is a dismantlable clique–Helly graph. Thus dually chordal graphs are properly contained in the classes of disk–Helly and clique–Helly graphs.

The paper is organized as follows. In section 2 we give standard hypergraph notions and properties. Section 3 is devoted to graphs with maximum neighborhood ordering. There we define some types of hypergraphs associated with graphs and present characterizations of dually chordal graphs, doubly chordal graphs, and strongly chordal graphs via hypergraph properties. The results of this section are from [20]. Section 4 deals with bipartite graphs with maximum neighborhood ordering. There we also describe relationships between graphs and bipartite graphs with different types of maximum neighborhood orderings. A part of the results of this section are from [6] and [22]. In section 5 some results confirming the duality between chordal graphs and dually chordal graphs are established. We conclude with two diagrams which present relationships between classes of graphs, hypergraphs, and some bipartite graphs.

2. Standard hypergraph notions and properties. We mainly use the hypergraph terminology of Berge [7]. A finite hypergraph $\mathcal{E}$ is a family of nonempty subsets (the *edges* of $\mathcal{E}$) from some finite underlying set V (the *vertices* of $\mathcal{E}$). The *subhypergraph* induced by a set $A \subseteq V$ is the hypergraph $\mathcal{E}_A$ defined on A by the edge set $\mathcal{E}_A = \{e \cap A : e \in \mathcal{E}\}$. The *dual hypergraph* $\mathcal{E}^*$ has $\mathcal{E}$ as its vertex set and $\{e \in \mathcal{E} : v \in e\}$ $(v \in V)$ as its edges. The 2-*section graph* $2SEC(\mathcal{E})$ of the hypergraph $\mathcal{E}$ has vertex set V, and two distinct vertices are adjacent if and only if they are contained in a common edge of $\mathcal{E}$. The *line graph* $L(\mathcal{E}) = (\mathcal{E}, E)$ of $\mathcal{E}$ is the intersection

graph of $\mathcal{E}$; i.e., $ee' \in E$ if and only if $e \cap e' \neq \emptyset$. A hypergraph $\mathcal{E}$ is *reduced* if no edge $e \in \mathcal{E}$ is contained in another edge of $\mathcal{E}$.

A hypergraph $\mathcal{E}$ is *conformal* if every clique C in $2SEC(\mathcal{E})$ is contained in an edge $e \in \mathcal{E}$. A *Helly hypergraph* is one whose edges satisfy the Helly property; i.e., any subfamily $\mathcal{E}' \subseteq \mathcal{E}$ of pairwise intersecting edges has a nonempty intersection.

First we give a list of well-known properties of hypergraphs (for these and other properties cf. [7]).

(i) Taking the dual of a hypergraph twice is isomorphic to the hypergraph itself; i.e., $(\mathcal{E}^*)^* \sim \mathcal{E}$.

(ii) $L(\mathcal{E}) \sim 2SEC(\mathcal{E}^*)$.

(iii) $\mathcal{E}$ is conformal if and only if $\mathcal{E}^*$ has the Helly property.

A hypergraph $\mathcal{E}$ is a *hypertree* (called *arboreal hypergraph* in [7]) if there is a tree T with vertex set V such that every edge $e \in \mathcal{E}$ induces a subtree in T (T is then called the *underlying vertex tree* of $\mathcal{E}$). A hypergraph $\mathcal{E}$ is a *dual hypertree* if there is a tree T with vertex set $\mathcal{E}$ such that for all vertices $v \in V$ $T_v = \{e \in \mathcal{E} : v \in e\}$ induces a subtree of T (T is then called the *underlying hyperedge tree* of $\mathcal{E}$).

Observe that $\mathcal{E}$ is a hypertree if and only if $\mathcal{E}^*$ is a dual hypertree.

A sequence $C = (e_1, e_2, \ldots, e_k, e_1)$ of edges is a *hypercycle* if $e_i \cap e_{i+1(mod\ k)} \neq \emptyset$ for $1 \leq i \leq k$. The *length* of C is k. A *chord* of the hypercycle C is an edge e with $e_i \cap e_{i+1(mod\ k)} \subseteq e$ for at least three indices i, $1 \leq i \leq k$. A hypergraph $\mathcal{E}$ is *α–acyclic* if it is conformal and contains no chordless hypercycles of length at least 3. Note that the notion of α–acyclicity was introduced in [5] in a different way but the notion given above is equivalent to that given in [5] (cf. [29]).

In a similar way, a graph G is *chordal* if it does not contain any induced (chordless) cycles of length at least 4.

THEOREM 1.

(i) *(See* [23], [27].*) $\mathcal{E}$ is a hypertree if and only if $\mathcal{E}$ is a Helly hypergraph and its line graph $L(\mathcal{E})$ is chordal.*

(ii) *(See* [5], [25], [29].*) $\mathcal{E}$ is a dual hypertree if and only if $\mathcal{E}$ is α–acyclic.*

Due to the dualities between hypertrees and dual hypertrees, the conformality and the Helly property, and the line graph of a hypergraph and the 2-section graph of the dual hypergraph, Theorem 1 can be expressed also in other variants by switching between a property and its dual.

A particular instance of hypertrees are totally balanced hypergraphs. A hypergraph is *totally balanced* if every cycle of length greater than two has an edge containing at least three vertices of the cycle.

THEOREM 2 (see [32]). *A hypergraph $\mathcal{E}$ is totally balanced if and only if every subhypergraph of $\mathcal{E}$ is a hypertree.*

There is a close connection between totally balanced hypergraphs, strongly chordal graphs and chordal bipartite graphs [1], [26], [11]; see [8] for a systematic treatment of these relations. Motivated by these results, we will establish similar connections between hypertrees, dually chordal graphs, and some classes of bipartite graphs.

Hypergraphs can be represented in a natural way by incidence matrices. Let $\mathcal{E} = \{e_1, \ldots, e_m\}$ be a hypergraph and $V = \{v_1, \ldots, v_n\}$ be its vertex set. The *incidence matrix* $\mathcal{IM}(\mathcal{E})$ of the hypergraph $\mathcal{E}$ is a matrix whose (i, j) entry is 1 if $v_i \in e_j$ and 0 otherwise. The (bipartite vertex–edge) *incidence graph* $\mathcal{IG}(\mathcal{E}) = (V, \mathcal{E}, E)$ of the hypergraph $\mathcal{E}$ is a bipartite graph with vertex set $V \cup \mathcal{E}$, where two vertices $v \in V$ and $e \in \mathcal{E}$ are adjacent if and only if $v \in e$. Note that the transposed matrix $\mathcal{IM}(\mathcal{E})^T$ is the incidence matrix of the dual hypergraph $\mathcal{E}^*$, while $\mathcal{IG}(\mathcal{E}) \sim \mathcal{IG}(\mathcal{E}^*)$ if the sides

of the bipartite graph are not marked.

Following [33] a matrix M is in *doubly lexical order* if rows and columns as 0-1-vectors are in increasing order. Two rows $r_1 < r_2$ and columns $c_1 < c_2$ form a Γ if the crossing points of these rows and columns define the submatrix $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$. An ordered 0-1 matrix M is *supported* Γ if for every pair $r_1 < r_2$ of rows and pair $c_1 < c_2$ of columns which form a Γ there is a row $r_3 > r_2$ with $M(r_3, c_1) = M(r_3, c_2) = 1$ (r_3 *supports* Γ).

A *subtree matrix* is the incidence matrix of a collection of subtrees of a tree T. A *totally balanced matrix* is the incidence matrix of a totally balanced hypergraph.

THEOREM 3. *Let M be a 0-1 matrix.*

(i) *(See* [33]*.) M is a subtree matrix if and only if it has a supported $\Gamma-$ordering.*

(ii) *(See* [1], [31], [33]*.) M is a totally balanced matrix if and only if it has a $\Gamma-$free ordering.*

Due to the duality shown later, part (i) of this theorem provides a matrix characterization of chordal graphs as well as dually chordal graphs by transposing the incidence matrix.

3. Maximum neighborhood orderings in graphs. Let $G = (V, E)$ be a finite undirected simple (i.e., without loops and multiple edges) and connected graph. For two vertices $x, y \in V$ the *distance* $d_G(x, y)$ is the length (i.e., number of edges) of a shortest path connecting x and y. Let $I(x, y) = \{v \in V : d_G(x, v) + d_G(v, y) = d_G(x, y)\}$ be the *interval* between vertices x and y. By $N_G(v) = \{u : uv \in E\}$ and $N_G[v] = N_G(v) \cup \{v\}$ we denote the *open neighborhood* and the *closed neighborhood* of v, respectively. If no confusion can arise we will omit the index G. Let $\mathcal{N}^0(G) = \{N(v) : v \in V\}$ and $\mathcal{N}(G) = \{N[v] : v \in V\}$ be the *open neighborhood hypergraph* and the *closed neighborhood hypergraph* of G, respectively. Let also $\mathcal{C}(G) = \{C : C$ is a maximal clique in $G\}$ be the *clique hypergraph* of G.

It is easy to see that the following holds:

(i) $2SEC(\mathcal{C}(G))$ is isomorphic to G (and thus $\mathcal{C}(G)$ is conformal).

(ii) $(\mathcal{N}(G))^*$ is isomorphic to $\mathcal{N}(G)$ (where it is assumed that the hypergraph $\mathcal{N}(G) = \{N[v] : v \in V\}$ is a multiset) and the same holds for $\mathcal{N}^0(G)$.

Concerning clique hypergraphs of chordal graphs, from Theorem 1 we have the following well-known equivalence:

(iii) A graph G is chordal if and only if its clique hypergraph $\mathcal{C}(G)$ is α–acyclic if and only if $\mathcal{C}(G)$ is a dual hypertree.

Let v be a vertex of G. The *disk* centered at v with radius k is the set of all vertices having distance at most k to v: $N^k[v] = \{u : u \in V$ and $d(u, v) \leq k\}$. Denote by $\mathcal{D}(G) = \{N^k[v] : v \in V$, k a positive integer$\}$ the *disk hypergraph* of G.

First we present some results establishing a connection between the closed neighborhood, the clique, and the disk hypergraphs of a given graph G.

Let a maximal induced cycle of G be an induced cycle of G with a maximum number of edges. Denote by $l(G)$ the number of edges of a maximal induced cycle of G.

LEMMA 1. *Let G be an arbitrary graph.*

(i) *$l(L(\mathcal{D}(G))) = l(L(\mathcal{N}(G)))$. In particular, $L(\mathcal{D}(G))$ is chordal if and only if $L(\mathcal{N}(G))$ is so.*

(ii) *$l(L(\mathcal{N}(G))) \leq l(L(\mathcal{C}(G)))$. In particular, if $L(\mathcal{C}(G))$ is chordal, then $L(\mathcal{N}(G))$ is so.*

(iii) *If $\mathcal{N}(G)$ is conformal, then $(\mathcal{C}(G))^*$ is so.*

Proof. (i) Among all maximal induced cycles of the graph $L(\mathcal{D}(G))$ choose a cycle

$$C = (N^{r_1}[v_1], \dots, N^{r_k}[v_k], N^{r_1}[v_1])$$

with a minimal sum $s = r_1 + \dots r_k$. We claim that C is formed by unit disks only; i.e., $r_1 = r_2 = \dots = r_k = 1$. Assume to the contrary that $r_1 \geq 2$.

Pick arbitrary vertices $a \in N^{r_1}[v_1] \cap N^{r_2}[v_2]$ and $b \in N^{r_k}[v_k] \cap N^{r_1}[v_1]$. Now consider two neighbors $v_1' \in I(a, v_1)$ and $v_1'' \in I(b, v_1)$ of the vertex v_1. If

$$N^{r_1-1}[v_1'] \cap N^{r_k}[v_k] = \emptyset, N^{r_1-1}[v_1''] \cap N^{r_2}[v_2] = \emptyset$$

holds then the disks

$$(N^{r_1-1}[v_1'], N^{r_2}[v_2], \dots, N^{r_k}[v_k], N^{r_1-1}[v_1''])$$

form an induced cycle with $k+1$ edges, contradicting the maximality of C.

So assume for example that $N^{r_1-1}[v_1'] \cap N^{r_k}[v_k] \neq \emptyset$. Then replacing the disk $N^{r_1}[v_1]$ by $N^{r_1-1}[v_1']$ in the cycle C we obtain an induced cycle with k edges and total radius sum $s-1$. This again contradicts the choice of C. Thus C consists of unit disks; i.e., C is an induced cycle of the graph $L(\mathcal{N}(G))$.

(ii) Consider vertices $v_1, \dots, v_k$ whose neighborhoods generate a maximal induced cycle C in the graph $L(\mathcal{N}(G))$. Let

$$B_2 = N[v_1] \cap N[v_2], \dots, B_k = N[v_{k-1}] \cap N[v_k], B_1 = N[v_k] \cap N[v_1].$$

In each set B_i pick a vertex b_i such that the sum $s = d(b_1, b_2) + \dots + d(b_{k-1}, b_k) + d(b_k, b_1)$ is minimal. Now define a cycle C' of the graph $L(\mathcal{C}(G))$ using the following rules: if the vertices $b_i, b_{i+1(mod\ k)}$ are adjacent, then add a clique K_i to C' which contains the vertices $b_i, b_{i+1(mod\ k)}$ and v_i; otherwise add two cliques K_i' and K_i'' (in this order) to C' which contain the edges v_ib_i and $v_ib_{i+1(mod\ k)}$, respectively.

The cycle C' has at least k edges. Assume that C' is not induced; i.e., two nonconsecutive cliques K' and K'' of C' have a nonempty intersection. By the definition of C' any clique of C' contains a center of some neighborhood from C. Since C is an induced cycle the cliques K' and K'' contain centers of two consecutive neighborhoods of C. Let us assume that $v_1 \in K'$ and $v_2 \in K''$. Up to symmetry we have one of the following possibilities: $K' = K_1$ and $K'' = K_2''$ or $K' = K_1'$ and $K'' = K_2'$ or $K' = K_1'$ and $K'' = K_2''$.

In all of these cases the inequality $d(b_1, b_2) + d(b_2, b_3) \geq 3$ holds. Let $b_2^* \in K' \cap K'' \subset B_2$. Since $d(b_1, b_2^*) + d(b_2^*, b_3) = 2$ this leads to a contradiction with the choice of the vertices $b_1, \dots, b_k$. Hence C' is an induced cycle of $L(\mathcal{C}(G))$ and its length is at least $k = l(L(\mathcal{N}(G)))$.

(iii) By the duality properties of hypergraphs it is sufficient to show that $\mathcal{C}(G)$ is a Helly hypergraph. Let $\mathcal{F} = \{C_1, \dots, C_m\}$ be a family of pairwise intersecting cliques. For each vertex $v \in \bigcup_{i=1}^m C_i$ consider the closed neighborhood $N[v]$. Evidently, any two such neighborhoods intersect. Therefore the vertices of the set $\bigcup_{i=1}^m C_i$ induce in $2SEC(\mathcal{N}(G))$ a clique. By the conformality of $\mathcal{N}(G)$ there exists a vertex w such that $N[w]$ contains the union $\bigcup_{i=1}^m C_i$. Due to the maximality of the cliques $C_1, \dots, C_m$ the vertex w belongs to all of them. □

3.1. Characterization of dually chordal graphs. Let $G = (V, E)$ be a graph. A vertex $v \in V$ is *simplicial* in G if $N[v]$ is a clique in G. Let $G_i = G(\{v_i, v_{i+1}, \dots, v_n\})$ be the subgraph induced by $\{v_i, v_{i+1}, \dots, v_n\}$ and $N_i[v]$ be the closed neighborhood

of v in G_i. A linear ordering $(v_1, \ldots, v_n)$ of V is a *perfect elimination ordering of* G if for all $i \in \{1, \ldots, n\}$, $N_i[v_i]$ is a clique; i.e., v_i is simplicial in G_i.

It is known that a graph G is chordal if and only if G has a perfect elimination ordering. Moreover, every noncomplete chordal graph has two nonadjacent simplicial vertices (see [28]).

A vertex $u \in N[v]$ is a *maximum neighbor of* v if for all $w \in N[v]$, $N[w] \subseteq N[u]$ holds (note that $u = v$ is not excluded). A linear ordering $(v_1, v_2, \ldots, v_n)$ of V is a *maximum neighborhood ordering* of G if for all $i \in \{1, \ldots, n\}$, there is a maximum neighbor $u_i \in N_i[v_i]$; i.e.,

$$\text{for all } w \in N_i[v_i], \quad N_i[w] \subseteq N_i[u_i] \text{ holds.}$$

Note that graphs with maximum neighborhood orderings are in general not perfect. Indeed, let $G = (V, E)$ be any graph and $x \notin V$ be a new vertex. Then for $G' = (V \cup \{x\}, E \cup \{vx : v \in V\})$ the ordering $(v_1, \ldots, v_n, x)$ is a maximum neighborhood ordering. Thus, e.g., the C_5 with an additional dominating vertex (the wheel W_5) has a maximum neighborhood ordering and is not perfect.

THEOREM 4. *For a graph G the following conditions are equivalent:*

(i) *G has a maximum neighborhood ordering*;
(ii) *there is a spanning tree T of G such that any maximal clique of G induces a subtree in T*;
(iii) *there is a spanning tree T of G such that any disk of G induces a subtree in T*;
(iv) *$\mathcal{N}(G)$ is a hypertree (is a dual hypertree).*

Proof. (i) $\Longrightarrow$ (ii). We proceed by induction on the number of vertices of the graph G. Let x be the first vertex in a maximum neighborhood ordering of G. Let y be a maximum neighbor of x; i.e., $N^2[x] = N[y]$. If $x = y$, then x is adjacent to all other vertices of G and the desired tree T could be a star with center x. Thus (ii) is fulfilled. Assume now that $x \neq y$. By induction hypothesis there exists a spanning tree of the graph $G - x = G(V \setminus \{x\})$ which satisfies condition (ii). Among all such spanning trees choose a tree T in which y is adjacent with a maximum number of vertices from $N(x)$. We claim that y is adjacent with all vertices from $N(x) \setminus \{y\}$.

Assume the contrary and pick a vertex $z \in N(x)$ which is nonadjacent to y in T. In T consider a path $y - \cdots - v - z$ connecting vertices y and z. Denote by T_v with $v \in T_v$ and T_z with $z \in T_z$ the connected components of T obtained by deleting an edge (v, z). Adding to these subtrees a new edge (y, z) we transform the tree T into a new tree T'. Since y and z are adjacent vertices of $G - x$ the tree T' is a spanning tree of $G - x$. Now we show that T' fulfills the condition (ii), too. Let C be a maximal clique of $G - x$. If $z \notin C$, then C is completely contained in one of the subtrees T_v or T_z; i.e., C induces in both trees T and T' one and the same subtree. So, suppose that $z \in C$. Since $N[z] \subseteq N[y] = N^2[x]$ we have $y \in C$. Let u_1, u_2 be arbitrary vertices from C. If both vertices u_1 and u_2 belong to one and the same subtree T_v or T_z, then these vertices are connected in T and T' by one and the same path, and we are done. Now, let $u_1 \in T_v$ and $u_2 \in T_z$. In T_v the vertices u_1 and y are connected by a path l_1, consisting of vertices from C. In a similar way, the vertices u_2 and z are joined in T_z by a path $l_2 \subseteq C$. Gluing together the paths l_1 and l_2 and the edge yz we obtain a path which connects the vertices u_1 and u_2 in T'. Hence any clique C of $G - x$ induces a subtree in T'; i.e., T' also satisfies condition (ii). This, however, contradicts the choice of the spanning tree T. The contradiction shows that y is adjacent in T to all vertices of $N(x) \setminus \{y\}$.

Consider a spanning tree T^* of G obtained from T by adding a leaf x adjacent to y. Evidently T^* fulfills condition (ii) of the theorem; i.e., T^* is the required tree.

(ii) $\Longrightarrow$ (iii). Let T be a spanning tree of G such that any clique of G induces a subtree in T. We claim that any disk $N^r[z]$ of G induces a subtree in T, too. In order to prove this, it is sufficient to show that the vertex z and any vertex $v \in N^r[z]$ may be connected in T by a path consisting of vertices from $N^r[z]$. Let $v = v_1 - v_2 - \cdots - v_k - v_{k+1} = z$ be a shortest path of G between v and z. By C_i we denote a maximal clique of G containing the edge $v_i v_{i+1}$, $i \in \{1, \ldots, k\}$. From the choice of T it follows that the vertices v_i and v_{i+1} are connected in T by a path $l_i \subseteq C_i$. The vertices of the set $L = \bigcup_{i=1}^{k} l_i$ induce a subtree $T(L)$ of the tree T. Therefore the vertices v and z may be connected in $T(L)$ (and in T, too) by a path l. Since $d(z, w) \leq d(z, v_i) \leq r$ for any vertex $w \in C_i$ any clique C_i belongs to the disk $N^r[z]$. So our claim follows from the following evident inclusions:

$$l \subseteq L \subseteq \bigcup_{i=1}^{k} C_i \subseteq N^r[z].$$

(iii) $\Longrightarrow$ (iv) is evident.

(iv) $\Longrightarrow$ (i). Suppose T is a tree with the same vertex set as G such that $N_G[v]$ induces a subtree T_v of T for all vertices v in G. Consider T as a tree rooted at a chosen vertex r. Every $N_G[v]$ has a unique vertex v^* such that

$$d_T(r, v^*) < d_T(r, u) \text{ for all vertices } u \in N_G[v] \setminus \{v^*\},$$

which can be considered as the *root* of the subtree T_v of T. Sort the vertices of G into $v_1, v_2, \ldots, v_n$ such that

$$d(r, v_1^*) \geq d(r, v_2^*) \geq \cdots \geq d(r, v_n^*).$$

We claim that this ordering is a maximum neighborhood ordering of G. Note that $v_i^* \in N_i[v_i]$. For each $v_j \in N_i[v_i]$ and $v_k \in N_i[v_j]$, v_j is in both T_{v_i} and T_{v_k}. So v_i^* and v_k^* are both ancestors of v_j. Also, $d_T(r, v_k^*) \leq d_T(r, v_i^*)$. Thus v_i^* is in the path from v_j to v_k^* in T. Since v_j and v_k^* are both in $N_G[v_k]$; i.e., in the subtree T_{v_k} of T, v_i^* is also in T_{v_k}; i.e., $v_i^* \in N_G[v_k]$ and so $v_k \in N_i[v_i^*]$. Thus v_i^* is a maximum neighbor of v_i for $1 \leq i \leq n$. This proves that $v_1, v_2, \ldots, v_n$ is a maximum neighborhood ordering of G. □

This result was also presented in [21].

In [40] a linear time algorithm for recognizing α–acyclicity of a hypergraph is given. Since dual hypertrees are exactly the α–acyclic hypergraphs by Theorem 4 we have the following.

COROLLARY 1. *It can be recognized in linear time $O(|V| + |E|)$ whether a graph G has a maximum neighborhood ordering.*

In [18], [9] we show that for a given dually chordal graph a maximum neighborhood ordering can be generated in linear time, too.

From Theorem 4 it also follows that G has a maximum neighborhood ordering if and only if $\mathcal{C}(G)$ is a hypertree. Recall that the graph G is chordal if and only if $(\mathcal{C}(G))^*$ is a hypertree. Thus graphs with maximum neighborhood ordering are dual to chordal graphs in this sense. Therefore we call them *dually chordal graphs*. The further results will confirm this term and will show the deepness of this duality. Note that unlike for chordal graphs where the number of maximal cliques is linearly bounded, this is not the case for dually chordal graphs.

Furthermore from Theorem 4 it follows that G has a maximum neighborhood ordering if and only if $\mathcal{D}(G)$ is a hypertree. Using this fact in [9] we present efficient algorithms for r–domination and r–packing problems on dually chordal graphs.

The *kth power* $G^k, k \geq 1$, of G has the same vertices as G, and two distinct vertices are joined by an edge in G^k if and only if their distance in G is at most k.

COROLLARY 2. *Any power of a dually chordal graph is dually chordal.*

Proof. Let G be a dually chordal graph, and let G^k be some power of this graph. A unit disk of G^k with center in v coincides with the disk $N^k[v]$ of G. Therefore $\mathcal{N}(G^k)$ is the family of all disks of radius k of the graph G. Since G is dually chordal $\mathcal{N}(G^k)$ has the Helly property and $L(\mathcal{N}(G^k))$ is chordal as an induced subgraph of the chordal graph $L(\mathcal{D}(G))$. By Theorem 4 it follows that G^k is dually chordal. □

3.2. Doubly chordal, power–chordal, and strongly chordal graphs. A vertex v of a graph G is *simple* [26] if the set $\{N[u] : u \in N[v]\}$ is totally ordered by inclusion. A linear ordering $(v_1, \ldots, v_n)$ of V is a *simple elimination ordering of* G if for all $i \in \{1, \ldots, n\}$ v_i is simple in G_i. A graph is *strongly chordal* if it admits a simple elimination ordering. A *k-sun* [11], [14], [26] is a graph with $2k$ vertices for some $k \geq 3$ whose vertex set can be partitioned into two sets $U = \{u_1, u_2, \ldots, u_k\}$ and $W = \{w_1, w_2, \ldots, w_k\}$ such that U induces a complete graph, W forms an independent set, and u_i is adjacent to w_j if and only if $i = j$ or $i = j + 1 (\text{mod } k)$.

COROLLARY 3. *For a graph G the following conditions are equivalent:*

(i) *G is a strongly chordal graph;*
(ii) *G is a sun-free chordal graph;*
(iii) *G is a hereditary dually chordal graph; i.e., any induced subgraph of G is dually chordal.*

Proof. The equivalence of (i) and (ii) is contained in [11], [14], [26]. Since every induced subgraph of a strongly chordal graph is strongly chordal we deduce that (i) $\Longrightarrow$ (iii). Furthermore, any simple vertex v of G evidently has a maximum neighbor. Finally (iii) $\Longrightarrow$ (ii) because induced cycles of length at least four and suns do not contain a vertex which has a maximum neighbor. □

By Lemma 1(iii) conformality of $\mathcal{N}(G)$ implies conformality of $(\mathcal{C}(G))^*$. Moreover in [16], [17] it has been shown that for chordal graphs $\mathcal{N}(G)$ is a Helly hypergraph if and only if $\mathcal{C}(G)$ is so. By Lemma 1(ii) we also know that $L(\mathcal{N}(G))$ is chordal if $L(\mathcal{C}(G))$ is chordal. The following result shows that for chordal graphs the converse is also true.

LEMMA 2. *For a chordal graph G the following conditions are equivalent:*

(i) *$G^2 \sim L(\mathcal{N}(G))$ is chordal;*
(ii) *$L(\mathcal{C}(G))$ is chordal.*

Proof. (ii) $\Longrightarrow$ (i) follows from Lemma 1(ii). Conversely, assume that there is an induced cycle $\Gamma = (C_1, \ldots C_m, C_1), m \geq 4$, of the graph $L(\mathcal{C}(G))$. Let $C = \bigcup_{i=1}^{m} C_i$. $G^2(C)$ as an induced subgraph of the chordal graph G^2 contains a simplicial vertex x. Suppose that $x \in C_1$. This means $C_2, C_m \subseteq N^2[x]$. Because of the simpliciality of x in G^2 for arbitrary vertices $u \in C_2$ and $v \in C_m$ we have $d(u, v) \leq 2$. Let $C_2 = \{x_1, \ldots x_s\}$ and $C_m = \{y_1, \ldots y_t\}$. We claim that any vertex of C_2 has in G a neighbor in C_m and vice versa. Assume to the contrary that this is not the case for x_1; i.e.,

$$d(x_1, y_1) = d(x_1, y_2) = \cdots = d(x_1, y_t) = 2.$$

Since G is chordal there exists a common neighbor of the vertices x_1 and $y_1, \ldots, y_t$. However, this contradicts the fact that C_m is a maximal clique of G. Thus our claim is true.

In the clique C_2 choose a vertex x_i which is adjacent to a maximum number of vertices from C_m. Suppose that x_i is adjacent to $y_1, \ldots, y_{l-1}$. Note that $l \leq t$,

otherwise $C_2 \cap C_m \neq \emptyset$. By our claim we conclude that y_l is adjacent to some vertex $x_j \in C_2$. A unique chord of the cycle $(x_i, y_k, y_l, x_j, x_i)$, $k \in \{1, \ldots, l-1\}$ may be only $x_j y_k$. Therefore x_j is adjacent with $y_1, \ldots, y_{l-1}, y_l$, contradicting the choice of x_i. So, our initial assumption that Γ is an induced cycle of $L(\mathcal{C}(G))$ leads to a contradiction. □

A graph is *power–chordal* if all of its powers are chordal. For the next theorem we need the following lemma.

LEMMA 3. *Let G be a noncomplete graph. If both graphs G and G^2 are chordal, then there exist two nonadjacent vertices of G which are simplicial in G and G^2.*

Proof. The assertion is evident when G^2 is complete. Assume that G^2 is noncomplete and let the assertion be true for all smaller graphs. Since G^2 is chordal there are two nonadjacent simplicial vertices in G^2. If both vertices are also simplicial in G, we are done. So, suppose that the simplicial vertex x of G^2 has in G two nonadjacent neighbors u and v. Consider a minimal $(u-v)$-separator F of the graph G. From [15] it follows that F is a complete subgraph of G. Evidently $x \in F$. Let $G(A)$ and $G(B)$ be connected components of $G(V \setminus F)$ containing u and v, respectively.

By the induction hypothesis either the subgraph $G_1 = G(A \cup F)$ contains a pair of two nonadjacent vertices which are simplicial in G_1 and G_1^2 or G_1 is a complete graph. In the first case at least one of the obtained vertices is in A (since F induces a complete subgraph). In the second case any vertex from A is simplicial in $G_1 = G_1^2$.

Summarizing we conclude that the set A contains a vertex y which is simplicial in G_1 and G_1^2. It is evident that y is simplicial in G. Now we show that y is simplicial in G^2, too. It is enough to consider only the case when y is adjacent in G with a vertex of F. For any vertex $w \notin A \cup F$ we have $d(w,x) \leq 2$ if $d(w,y) \leq 2$. Since y is simplicial in G_1^2 a similar implication also holds for any vertex $u \in A \cup F$: if $d(u,y) \leq 2$, then $d(u,x) \leq 2$. Hence for arbitrary vertices v and w such that $d(y,w) \leq 2$ and $d(y,v) \leq 2$ we have analogous inequalities $d(x,w) \leq 2$ and $d(x,v) \leq 2$. Now, recall that x is simplicial in G^2. This implies that $d(v,w) \leq 2$ and y is simplicial in G^2.

In a similar way we obtain the existence of a vertex $z \in B$ which is simplicial in G and G^2. It remains to notice that y and z are nonadjacent. □

THEOREM 5. *For a graph G the following conditions are equivalent:*

(i) *G is power–chordal;*
(ii) *G and G^2 are chordal;*
(iii) *there exists a common perfect elimination ordering of G and G^2 (i.e., an ordering$(v_1, \ldots, v_n)$ of V such that v_i is simplicial in both graphs G_i and G_i^2, $i \in \{1, \ldots, n\}$).*

Proof. In [24] it is shown that if G^k is chordal, then so is G^{k+2}. Consequently, powers of chordal graphs are chordal provided that G^2 is chordal; i.e., (i) $\Longleftrightarrow$ (ii). The implication (iii) $\Longrightarrow$ (ii) is evident. To prove that (ii) $\Longrightarrow$ (iii) we proceed by induction on the number of vertices. By Lemma 3 there is a simplicial vertex v of G and G^2. It is easy to see that $(G-v)^2 = G^2 - v$; i.e., both graphs $G-v$ and $(G-v)^2$ are chordal. Applying to these graphs the induction hypothesis we obtain the required common perfect elimination ordering. □

A vertex v of a graph G is *doubly simplicial* [34] if v is simplicial and has a maximum neighbor. A linear ordering $(v_1, \ldots, v_n)$ of the vertices of G is *doubly perfect* if for all $i \in \{1, \ldots, n\}$ v_i is a doubly simplicial vertex of G_i. A graph G is *doubly chordal* [34] if it admits a doubly perfect ordering. The following result justifies the term "doubly chordal graphs."

COROLLARY 4 (See [20], [34]). *For a graph G the following conditions are equiv-*

alent:

(i) *G is doubly chordal*;

(ii) *G is chordal and dually chordal*;

(iii) *both hypergraphs $\mathcal{C}(G)$ and $(\mathcal{C}(G))^*$ are hypertrees.*

Proof. From the previous results it is sufficient to show that (ii) $\Longrightarrow$ (i). Since G and G^2 are chordal, Theorem 5 ensures the existence of a vertex v which is simplicial in G and G^2. For any two vertices $x, y \in N^2[v]$ the inequality $d(x,y) \leq 2$ is fulfilled. Hence $N[x] \cap N[y] \neq \emptyset$. Since $\mathcal{N}(G)$ is a hypertree the family of pairwise intersecting disks $\{N[x] : x \in N^2[v]\}$ has a nonempty intersection. Let w be a vertex from this intersection. Then w is a maximum neighbor of v. As we already mentioned $(G-v)^2 = G^2 - v$. It remains to show that $\mathcal{N}(G-v)$ has the Helly property. But this is obvious, because any neighborhood containing v contains the vertex w, too. □

From these results we conclude that powers of doubly chordal graphs are doubly chordal. For strongly chordal graphs a similar result was established in [33]: Powers of strongly chordal graphs are strongly chordal.

4. Maximum neighborhood orderings in bipartite graphs. Let $G = (V, E)$ be an arbitrary graph, and let v be a vertex of G. Following [3] the sets

$$HD_{odd}(v) = \{u \in V : d(u,v) \leq k \text{ and } d(u,v) \text{ is odd}\},$$

$$HD_{even}(v) = \{u \in V : d(u,v) \leq k \text{ and } d(u,v) \text{ is even}\}$$

are called the *half-disks* centered at v with radius k. By $\mathcal{HD}(G)$ we denote the family of all half-disks of G and call it the *half–disk hypergraph* of the graph G.

4.1. Bipartite graphs with maximum X–neighborhood ordering. For bipartite graphs $B = (X, Y, E)$ there are also standard hypergraph constructions: $\mathcal{N}^X(B) = \{N(y) : y \in Y\}$ denotes the *X–sided neighborhood hypergraph* of B (analogously define $\mathcal{N}^Y(B)$). Note that $(\mathcal{N}^X(B))^*$ is isomorphic to $\mathcal{N}^Y(B)$ and the same for X and Y exchanged. In addition, $\mathcal{N}^0(B) = \mathcal{N}^X(B) \cup \mathcal{N}^Y(B)$.

The half-disks of a bipartite graph B are defined as follows: for $z \in X$ let $HD_B^X(z,k) = \{x : x \in X \text{ and } d(z,x) \leq k \text{ and } d(z,x) \text{ even}\}$ and for $z \in Y$ let $HD_B^X(z,k) = \{x : x \in X \text{ and } d(z,x) \leq k \text{ and } d(z,x) \text{ odd}\}$ (the half-disks in X). Analogously define the half-disks in Y. Again if no confusion can arise we will omit the index B. The half–disk hypergraph $\mathcal{HD}(B)$ of the bipartite graph B splits into two components: $\mathcal{HD}^X(B) = \{HD^X(y, 2k+1) : y \in Y \text{ and } k \text{ a positive integer}\} \cup \{HD^X(x, 2k) : x \in X \text{ and } k \text{ a positive integer}\}$, called the *$X$–sided half–disk hypergraph* (consisting of subsets of X), and $\mathcal{HD}^Y(B)$ (defined analogously) called the *Y–sided half–disk hypergraph* (consisting of subsets of Y); i.e., $\mathcal{HD}(B) = \mathcal{HD}^X(B) \cup \mathcal{HD}^Y(B)$.

A bipartite graph $B = (X, Y, E)$ is called *X–conformal* [2] if for any set $S \subseteq Y$ with the property that all vertices of S have pairwise distance 2 there is a vertex $x \in X$ with $S \subseteq N(x)$. B is *X–chordal* [2] if for every cycle C in B of length at least 8 there is a vertex $x \in X$ which is adjacent to at least two vertices in C whose distance in C is at least 4 (a *bridge vertex*). Analogously define Y–chordality and Y–conformality. In [2] it is also shown that the following connection holds.

LEMMA 4. *Let $B = (X, Y, E)$ be a bipartite graph. Then B is X–chordal and X–conformal if and only if $\mathcal{N}^Y(B)$ is a dual hypertree if and only if $\mathcal{N}^X(B)$ is a hypertree.*

A vertex $y \in N(x)$ of $B = (X, Y, E)$ is a *maximum neighbor of* x if for all $y' \in N(x)$ $N(y') \subseteq N(y)$ holds. Let $B_i^Y = B(X \cup \{y_i, y_{i+1}, \ldots, y_n\})$ and $N_i(x)$ be the neighborhood of $x \in X$ in B_i^Y. A linear ordering $(y_1, \ldots, y_n)$ of Y is a *maximum X–neighborhood ordering* of B if for all $i \in \{1, \ldots, n\}$ there is a maximum neighbor $x_i \in N(y_i)$ of y_i; i.e.,

$$\text{for all } x \in N(y_i) \quad N_i(x) \subseteq N_i(x_i) \text{ holds.}$$

Analogously define a *maximum Y–neighborhood ordering.*

THEOREM 6. *Let $B = (X, Y, E)$ be a bipartite graph. Then the following conditions are equivalent:*

(i) *B has a maximum X–neighborhood ordering*;
(ii) *B is X–chordal and X–conformal*;
(iii) *$\mathcal{N}^X(B)$ is a hypertree*;
(iv) *the X–sided half-disk hypergraph $\mathcal{HD}^X(B)$ is a hypertree.*

Proof. The equivalence (ii) $\Longleftrightarrow$ (iii) follows from Lemma 4. The direction (iv) $\Longrightarrow$ (iii) is obvious.

(i) $\Longrightarrow$ (ii). Let $(y_1, \ldots, y_n)$ be a maximum X–neighborhood ordering of Y. Consider a chordless cycle $C = (x_{i_1}, y_{i_1}, \ldots, x_{i_k}, y_{i_k})$, $k \geq 4$. Assume that y_{i_1} is the leftmost Y–vertex of C in $(y_1, \ldots, y_n)$ which appears in this ordering in the jth position: $y_{i_1} = y_j$. Since $y_{i_k} \in N_j(x_{i_1}) \setminus N_j(x_{i_2})$ and $y_{i_2} \in N_j(x_{i_2}) \setminus N_j(x_{i_1})$ the sets $N_j(x_{i_1})$ and $N_j(x_{i_2})$ are incomparable with respect to set inclusion. Thus neither x_{i_1} nor x_{i_2} are maximum neighbors of y_{i_1}. Let x be a maximum neighbor of $y_{i_1} = y_j$. Then $y_{i_1}, y_{i_2}, y_{i_k} \in N_j(x)$ and x is a bridge vertex. (Note that x is even a neighbor of three Y–vertices of C.) Thus B is X-chordal.

Now let $S \subseteq Y$ be a subset of vertices of pairwise distance 2. Let $y \in S$ be the leftmost element of S in $(y_1, \ldots, y_n)$ and assume that $y = y_j$. For all $y' \in S$ there are common neighbors $x' \in X$ of y and y'. If x is a maximum neighbor of y_j, then $S \subseteq N_j(x)$. Thus B is X–conformal.

(ii) $\Longrightarrow$ (i). Assume that B is X–chordal and X–conformal. By Lemma 4 the graph $G' = 2SEC(\mathcal{N}^Y(B))$ is chordal. Let $(y_1, \ldots, y_n)$ be a perfect elimination ordering of G'. Thus $N_{G'}[y_1]$ is a clique; i.e., for all $u, v \in N_{G'}[y_1]$, $u \neq v$, there is a common neighbor in X and so the distance between u and v is 2. Since B is X–conformal there is an $x \in X$ with $N_{G'}[y_1] \subseteq N_B(x)$. Necessarily x is a neighbor of y_1 in B and is also a maximum neighbor of y_1 in B since for all $x' \in X$ with $x' \in N_B[y_1]$ $N_B(x') \subseteq N_{G'}[y_1]$.

The same argument can be applied repeatedly to the graph B_i^Y since $G' \setminus \{y_1\}$ is again chordal. Thus the perfect elimination ordering $(y_1, \ldots, y_n)$ of G' is a maximum X–neighborhood ordering of B and vice versa.

(iii) $\Longrightarrow$ (iv). Suppose that T_N is a tree with vertex set X such that for all $y_i \in Y$, $i \in \{1, \ldots, n\}$, $N(y_i)$ induces a subtree in T_N, and let $(y_1, \ldots, y_n)$ be a maximum X–neighborhood ordering of Y. We have to show that then also each half–disk of $\mathcal{HD}^X(B)$ induces a subtree in T_N, too.

The proof is done along the maximum X–neighborhood ordering $(y_1, \ldots, y_n)$ of Y. Let Y_i denotes the subset $\{y_i, \ldots, y_n\}$, and let B_i be the bipartite graph B restricted to Y_i. For Y_n the assertion is obviously true since the only X–sided half-disks in this case are the one-vertex sets $\{x\}, x \in X$, and the neighborhood $N(y_n)$. Obviously, these sets induce subtrees of T_N. Assume now that the half-disks of $\mathcal{HD}^X(B_{i+1})$ induce subtrees in T_N, $i \geq 1$. We will show that then also the half-disks of $\mathcal{HD}^X(B_i)$ induce subtrees in T_N. Without loss of generality let $i = 1$. Let x be a maximum

neighbor of y_1. In order to show that the half-disks of $\mathcal{HD}^X(B)$ induce subtrees of T_N we describe their structure. Consider for example the half–disk centered at z with radius $k \geq 2$. We distinguish two cases.

Case 1. $z \in N(y_1)$. First suppose that the degree of z is 1; i.e., $N(z) = \{y_1\}$.

Then $HD_B^X(z,k) = HD_{B_2}^X(x,k-2) \cup N(y_1)$ as can be easily seen: for every vertex $w \notin N(y_1)$ we have $d(x,w) = d(z,w) - 2$ and thus $w \in HD_{B_2}^X(x,k-2)$. Otherwise, if the degree of z is larger than 1, then $HD_B^X(z,k) = HD_{B_2}^X(z,k) \cup N(y_1)$; indeed, for every vertex $w \notin N(y_1)$ there is a path of length $d(z,w)$ which avoids the vertex y_1.

Case 2. $z \notin N(y_1)$.

If $d(z,y_1) > k$, then $HD_B^X(z,k) = HD_{B_2}^X(z,k)$. Otherwise, we obtain that $HD_B^X(z,k) = HD_{B_2}^X(z,k) \cup N(y_1)$. Furthermore in the latter case the vertex x belongs to $HD_{B_2}^X(z,k)$; as $HD_{B_2}^X(z,k)$ contains a neighbor x_j of y_1 and since x is a maximum neighbor of y_1 the half–disk also contains x itself.

Thus in all cases either $HD_B^X(z,k)$ is the same as before or is a union of two subtrees of T_N which both contain x. Thus it is again a subtree of T_N. □

From the proof of the implication (ii) $\Longrightarrow$ (i) it follows also that $(y_1, \ldots, y_n)$ is a maximum X–neighborhood ordering of B if and only if $(y_1, \ldots, y_n)$ is a perfect elimination ordering of $2\mathrm{SEC}(\mathcal{N}^Y(B))$.

4.2. Graphs with *b*–extremal ordering. Now let G be again an arbitrary graph. Lemma 1 gives a connection between the closed neighborhood and the disk hypergraphs of a given graph G. The next lemma establishes a similar connection between the open neighborhood hypergraph and the half–disk hypergraph of a graph G.

LEMMA 5. *For any graph G $l(L(\mathcal{HD}(G))) = l(L(\mathcal{N}^0(G)))$ holds. In particular, $L(\mathcal{HD}(G))$ is chordal if and only if $L(\mathcal{N}^0(G))$ is so.*

In a graph G a vertex v is *dominated* by another vertex $u \neq v$ if $N(v) \subseteq N(u)$. A vertex v is *b–extremal* if it is dominated by another vertex and there exists a vertex w such that $N(N(v)) = N(w)$. The ordering $(v_1, \ldots, v_n)$ of V is a *b–extremal ordering* of G if for all $i \in \{1, \ldots, n\}$ v_i is b–extremal in G_i. It is quite evident that a graph G admitting a b–extremal ordering must be bipartite. Indeed, consider the following iterative coloring of G. Let the vertices v_n and v_{n-1} be colored. Then for any i $(i < n-1)$ if the vertex v_i is dominated by v_j, then v_i gets the same color as v_j.

THEOREM 7. *For a graph G the following conditions are equivalent:*

(i) *$\mathcal{N}^0(G)$ is a hypertree*;
(ii) *$\mathcal{HD}(G)$ is a hypertree*;
(iii) *G is bipartite, and G has a maximum X–neighborhood ordering and a maximum Y–neighborhood ordering*;
(iv) *G has a b–extremal ordering.*

Proof. The equivalence of (i), (ii), and (iii) is an immediate consequence of Theorem 6 and the fact that if $\mathcal{N}^0(G)$ is a hypertree, then G is bipartite (which has a straightforward proof).

(i) $\Longrightarrow$ (iv). Let $\mathcal{N}^0(G)$ be a hypertree. Then G is bipartite, say, $G = (X, Y, E)$. Consider the chordal graphs $G^Y = 2SEC(\mathcal{N}^Y(G))$ and $G^X = 2SEC(\mathcal{N}^X(G))$. Let $x \in X$ be a simplicial vertex of G^X. Additionally suppose that x is an opposite vertex in G for some v; i.e., $x \notin I(v,x')$ for any vertex x' of G. Since x is simplicial in G^X the distance between every two vertices from $N(N(x))$ is 2. By the Helly property there is a vertex $y \in Y$ such that $N(y) = N(N(x))$. Moreover, since x is an opposite vertex for v and G is bipartite, necessarily $N(x) \subseteq I(x,v)$. Consider the family of half-disks consisting of open neighborhoods centered at vertices of $N(x)$ and a half-disk centered

at v with radius $d(x,v)-1$. By the Helly property for half-disks there is a common vertex $z \neq x$ of these open neighborhoods. So $N(x) \subseteq N(z)$. Hence any vertex x, which is opposite in G and simplicial in G^X or G^Y, is a b-extremal vertex of the graph G.

Now we prove that such a vertex always exists. Let $\mathrm{diam}(G)$, $\mathrm{diam}(G^X)$, and $\mathrm{diam}(G^Y)$ be the diameters of the graphs G, G^X, and G^Y, respectively. First assume that $\mathrm{diam}(G)$ is even; i.e., $\mathrm{diam}(G)=2k$. Then $\max(\mathrm{diam}(G^X), \mathrm{diam}(G^Y))=k$. Let $\mathrm{diam}(G^X)=k$ and let x' and x'' be a diametral pair in G^X. Then $d(x',x'') = 2k = \mathrm{diam}(G)$; i.e., any diametral pair of G^X is a diametral pair of G, too. It is known [41] that any chordal graph contains a diametral pair of simplicial vertices. Now assume that $\mathrm{diam}(G)$ is odd; i.e., $\mathrm{diam}(G)=2k+1$. Then $\max(\mathrm{diam}(G^X), \mathrm{diam}(G^Y))=k$. Let $\mathrm{diam}(G^X)=k$, and let x' and x'' be simplicial vertices which constitute a diametral pair of G^X. Then either x' and x'' are mutually opposite vertices in G or one of them is an end of a diameter in G.

(iv) $\Longrightarrow$ (iii). If $G=(V,E)$ has a b–extremal ordering $(v_1,\ldots,v_n)$, then by arguments above G is bipartite: $G=(X,Y,E)$. Assume that $v_1 \in Y$. Let $G' = G - v_1 = (X, Y - v_1, E')$ with a maximum Y–neighborhood ordering $(x_1,\ldots,x_r)$ and a maximum X–neighborhood ordering $(y_1,\ldots,y_s)$. Then $(v_1,y_1,\ldots,y_s)$ is also a maximum X–neighborhood ordering of G: it is obvious that v_1 has a maximum neighbor in X. Furthermore, as we will show, $(x_1,\ldots,x_r)$ is still a maximum Y–neighborhood ordering of G. Assume by way of contradiction that for x_1 this is not so. Let z be a maximum neighbor of x_1 in G', and assume that $N(z)$ and $N(v_1)$ are incomparable with respect to set inclusion. Since v_1 is b–extremal there is a vertex $u \in Y \setminus \{v_1\}$ such that $N(v_1) \subseteq N(u)$, contrary to the maximality of $N(z)$ in G'. □

Recall [30] that a graph G is *chordal bipartite* if G is bipartite and any induced cycle of G has length 4.

Corollary 5. *For a graph the following conditions are equivalent:*

(i) *Every induced subgraph of G admits a b–extremal ordering*;

(ii) *G is a chordal bipartite graph.*

We conclude this section by establishing some relationships between dually chordal graphs and their bipartite relatives. For this we recall two standard transformations of graphs. The first transformation associates with a graph $G=(V,E)$ the bipartite graph $B(G)$, called the *bigraph* of G. The vertex set of $B(G)$ consists of two disjoint copies V' and V'' of V, with $v' \in V'$ and $w'' \in V''$ adjacent in $B(G)$ if and only if v and w either coincide or are adjacent in G. Equivalently, $B(G)$ is the (vertex–closed-neighborhood) incidence graph of G; i.e., $B(G) = \mathcal{IG}(\mathcal{N}(G))$. In a similar way we define the bipartite graph $B_C(G) = \mathcal{IG}(\mathcal{C}(G))$.

From Theorems 4, 6, and 7 we obtain the following result.

Corollary 6. *Let G be a graph. Then G has a maximum neighborhood ordering if and only if $B(G)$ has a maximum X–neighborhood ordering (maximum Y–neighborhood ordering) if and only if $B(G)$ has a b–extremal ordering.*

Let $B=(X,Y,E)$ be a bipartite graph. Then the graph $\mathrm{split}_X(B)=(X \cup Y, E_X)$ is obtained from B by completing X to a clique. Assume that X is a maximal clique in $\mathrm{split}_X(B)$, i.e., for no $y \in Y$ $X \subseteq N(y)$. Note that the set of maximal cliques in $\mathrm{split}_X(B)$ is

$$\mathcal{C}(\mathrm{split}_X(B)) = \{\{y\} \cup N(y) : y \in Y\} \cup \{X\}.$$

Lemma 6. *Let $B=(X,Y,E)$ be a bipartite graph.*

(i) *$\mathcal{N}^X(B)$ has the Helly property if and only if $\mathcal{C}(\mathrm{split}_X(B))$ has the Helly property (analogously for Y instead of X).*
(ii) *$L(\mathcal{N}^X(B))$ is chordal if and only if $L(\mathcal{C}(\mathrm{split}_X(B)))$ is chordal.*

Thus $\mathcal{N}^X(B)$ is a hypertree if and only if $\mathcal{C}(\mathrm{split}_X(B))$ is a hypertree.

The assertion (i) follows from the definition of $\mathcal{N}^X(B)$ and $\mathcal{C}(\mathrm{split}_X(B))$. To show (ii) let $L = L(\mathcal{N}^X(B)) = (\{N(y) : y \in Y\}, E')$ and $(N(y_1), \ldots, N(y_k))$ be a perfect elimination ordering of L. Then $N(y_1)$ is a simplicial vertex in L; i.e., all $N(y)$ intersecting $N(y_1)$ are pairwise intersecting (the elements in the intersection are elements of X). Then for $R = L(\mathcal{C}(\mathrm{split}_X(B)))$ $(N[y_1], \ldots, N[y_k], X)$ is a perfect elimination ordering of R and vice versa.

COROLLARY 7. *Let $B = (X, Y, E)$ be a bipartite graph. Then B is X–chordal and X–conformal if and only if $\mathrm{split}_X(B)$ is doubly chordal.*

The proof of this result is a sequence of equivalences: B is X–chordal and X–conformal if and only if $\mathcal{N}^X(B)$ is a hypertree if and only if $\mathcal{C}(\mathrm{split}_X(B))$ is a hypertree if and only if $\mathrm{split}_X(B)$ is doubly chordal.

In section 2 we gave the notion of the incidence graph $\mathcal{IG}(\mathcal{E})$ of a hypergraph $\mathcal{E}$. In the particular case of one-sided neighborhood hypergraphs $\mathcal{N}^V(\mathcal{IG}(\mathcal{E}))) = \mathcal{E}$ and $\mathcal{N}^{\mathcal{E}}(\mathcal{IG}(\mathcal{E}))) = \mathcal{E}^*$ hold.

COROLLARY 8. *Let $\mathcal{E}$ be a hypergraph. Then $\mathcal{E}$ is a hypertree if and only if $\mathcal{IG}(\mathcal{E})$ has a maximum X–neighborhood ordering if and only if $\mathrm{split}_V(\mathcal{IG}(\mathcal{E}))$ has a maximum neighborhood ordering.*

5. The duality between chordal and dually chordal graphs. In this section we take advantage of the previous results to explain the duality between chordal and dually chordal graphs.

THEOREM 8. *Let $G = (V, E)$ be a graph.*
(i) *G is chordal if and only if $B_C(G)$ has a maximum y–neighborhood ordering.*
(ii) *G is dually chordal if and only if $B_C(G)$ has a maximum X–neighborhood ordering.*
(iii) *G is doubly chordal if and only if $B_C(G)$ has a X–neighborhood ordering and a maximum Y–neighborhood ordering if and only if $B_C(G)$ has a b–extremal ordering.*

It is well known [12] that chordal graphs are exactly the intersection graphs of subtrees of a tree. The next result shows that a dual property characterizes the class of dually chordal graphs.

THEOREM 9. *Let $G = (V, E)$ be a graph.*
(i) *(See [12]) G is chordal if and only if it is the line graph of some hypertree if and only if it is the 2–section graph of some α–acyclic hypergraph.*
(ii) *G is dually chordal if and only if it is the line graph of some α–acyclic hypergraph if and only if it is the 2–section graph of some hypertree if and only if it is the 2–section graph of paths of a tree.*
(iii) *G is doubly chordal if and only if it is the line graph of some α–acyclic hypertree if and only if it is the 2–section graph of some α–acyclic hypertree.*

Proof. To show (ii) let G be a dually chordal graph. By Theorem 4 $\mathcal{E} = \mathcal{C}(G)$ is a hypertree. Recall also that $G = 2SEC(\mathcal{C}(G)) = 2SEC(\mathcal{E})$. Let T be a representing tree of $\mathcal{E}$. We obtain the hypergraph $\mathcal{E}'$ of paths of the tree T from $\mathcal{E}$ by replacing every subtree T_C $(C \in \mathcal{C}(G))$ by a collection of all paths connecting in T the leaves of T_C. Obviously, $2SEC(\mathcal{E})$=$2SEC(\mathcal{E}')$.

Now assume that G is the 2–section graph of some hypertree $\mathcal{E}$ with representing tree T. Consider a neighborhood $N[v]$ in G. Since $N[v]$ is a union of subtrees con-

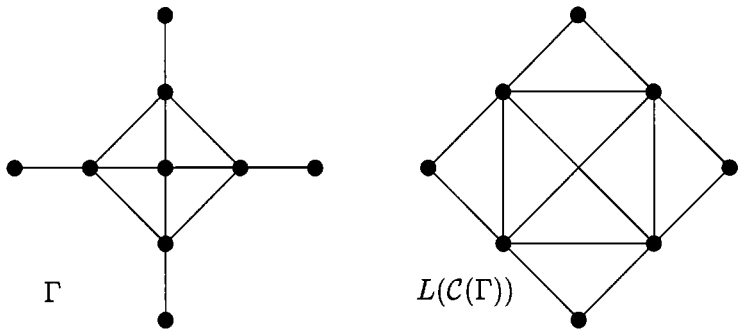


FIG. 1.

taining v, $N[v]$ is a subtree of T; i.e., $\mathcal{N}(G)$ is a hypertree. By Theorem 4 G is dually chordal. □

It is well known that there is a simple method to obtain an underlying hyperedge tree for the clique hypergraph $\mathcal{C}(G)$ of a chordal graph G (the so-called *clique–tree* [37]): Weight the edges of the intersection graph $L(\mathcal{C}(G))$ by the size of the intersection and find a maximum spanning tree on this graph.

For a dually chordal graph G there is a dual variant of this method (see [18]): Weight every edge of the graph $G = 2SEC(\mathcal{C}(G))$ by the number of maximal cliques of G containing this edge and find a maximum spanning tree on this weighted graph. Then G is dually chordal if and only if every maximum spanning tree on G is an underlying vertex tree for $\mathcal{C}(G)$.

As it was shown in [33] the matrix M^TM (M^T the transpose of M) is totally balanced provided that M is so. Unfortunately a similar property does not hold for subtree matrices; see Figure 1. The graph Γ is dually chordal. So the incidence matrix $M = \mathcal{IM}(\mathcal{C}(\Gamma))$ is a subtree matrix. The matrix M^TM is the neighborhood matrix $M = \mathcal{IM}(\mathcal{N}(L(\mathcal{C}(\Gamma))))$ of the clique graph $L(\mathcal{C}(\Gamma))$ of Γ. Since $L(\mathcal{C}(\Gamma))$ is not dually chordal M^TM is not a subtree matrix. Nevertheless the following is true.

COROLLARY 9. *If M is a subtree matrix then so is MM^T.*

Proof. Let $\mathcal{E}_M$ be a hypertree whose incidence matrix is M. By Theorem 9 the graph $G = 2SEC(\mathcal{E}_M)$ is dually chordal. Note that the matrix MM^T is the neighborhood matrix $\mathcal{IM}(\mathcal{N}(G))$. Since $\mathcal{N}(G)$ is a hypertree (Theorem 4) MM^T is a subtree matrix. □

The graph Γ of Figure 1 shows that the clique graph of a dually chordal graph is not necessarily dually chordal. The results below characterize the clique graphs of chordal, dually chordal, and doubly chordal graphs.

Subsequently we use the following notations: A graph G is *clique–Helly* if $\mathcal{C}(G)$ has the Helly property. G is *Helly chordal* if G is chordal and clique–Helly. G is *clique–chordal* if $L(\mathcal{C}(G))$ is chordal.

COROLLARY 10. *G is a Helly chordal graph if and only if G is the clique graph of some dually chordal graph G'; i.e., $G \sim L(\mathcal{C}(G'))$.*

Proof. By Theorem 4 the clique hypergraph $\mathcal{C}(G')$ has the Helly property. By Theorem 9 $L(\mathcal{C}(G'))$ is chordal. On the other hand, as follows from [4, Theorem 3.2], cliques of the graph $L(\mathcal{C}(G'))$ have the Helly property. Conversely, assume that G is a Helly chordal graph. By Theorem 9 G is the line graph of some conformal hypertree $\mathcal{E}$. It is easy to see that any conformal and reduced hypergraph is the

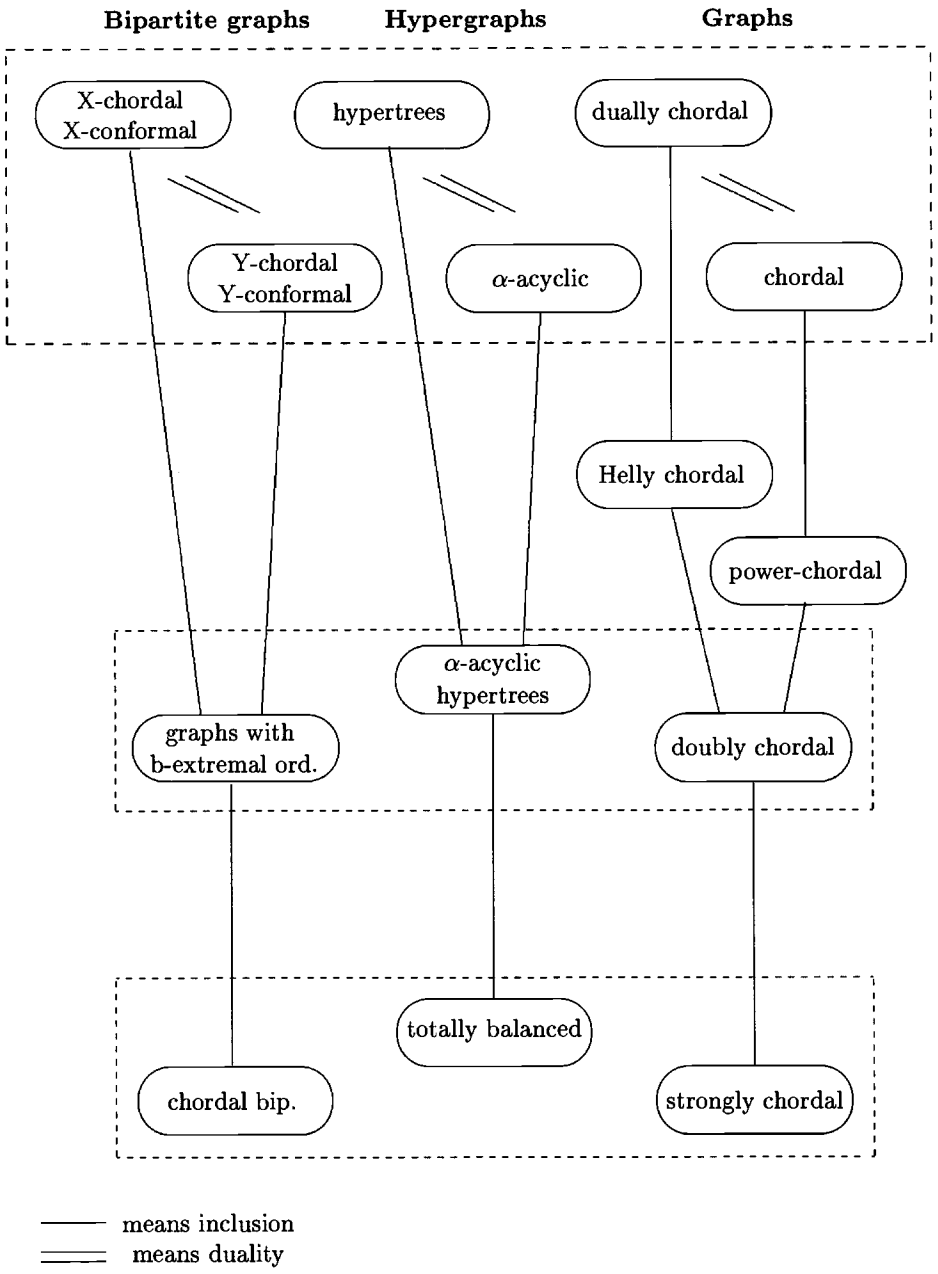


FIG. 2.

clique hypergraph of its 2–section graph. Then any conformal and reduced hypertree is the clique hypergraph of some dually chordal graph. So it is sufficient to transform $\mathcal{E}$ into such a hypergraph $\mathcal{E}'$ without changing its line graph. We obtain the hypergraph $\mathcal{E}'$ from $\mathcal{E}$ by adding to each edge e_i of $\mathcal{E}$ one new vertex u_i incident to e_i only. □

COROLLARY 11 (see [39]). *G is a dually chordal graph if and only if G is the clique graph of some chordal graph if and only if G is the clique graph of some intersection*

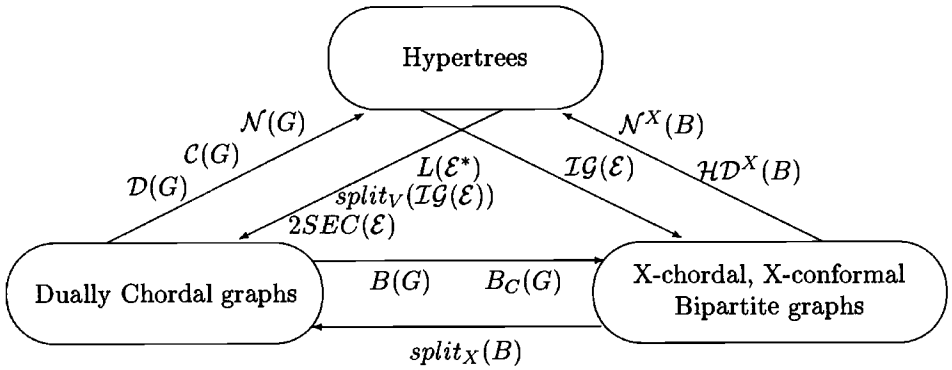


FIG. 3.

graph of paths in a tree.

The proof follows from Theorem 9 by using similar arguments as in the proof of Corollary 10.

Combining Corollaries 10 and 11 and Theorem 9 we obtain the following.

COROLLARY 12. *G is a doubly chordal graph if and only if G is the clique graph of some doubly chordal graph.*

Our duality results are established using the clique hypergraph $\mathcal{C}(G)$ of a graph G. The following four properties of this hypergraph play a crucial role:

- Conformality of $\mathcal{C}(G)$;
- Chordality of $G = 2SEC(\mathcal{C}(G))$;
- Helly property of $\mathcal{C}(G)$;
- Chordality of $L(\mathcal{C}(G))$.

The conformality of $\mathcal{C}(G)$ is fulfilled for all graphs. Chordal graphs are a well–investigated class; see, for instance, [28]. Clique–Helly graphs are characterized in [4], [16], [17].

In different combinations the four conditions above characterize the graph classes considered in this paper.

dually chordal = **clique–Helly** $\bigcap$ **clique–chordal**
doubly chordal = **clique–Helly** $\bigcap$ **clique–chordal** $\bigcap$ **chordal**
Helly chordal = **clique–Helly** $\bigcap$ **chordal**
Power–chordal = **clique–chordal** $\bigcap$ **chordal**

We conclude with the hint to two diagrams (Figures 2 and 3) which show the relations between graph classes and hypergraphs associated with these graphs.

6. Concluding remarks. We have shown the close relationship of graphs with maximum neighborhood ordering and hypergraph properties as the Helly property and tree-like representations of maximal cliques and neighborhoods. Thus in the sense of hypergraph duality these graphs are dual to chordal graphs but have different properties, especially they are in general not perfect. On the other hand maximum neighborhood orderings turn out to be very useful for domination-like problems (see [21], [34], [18], [6], [19]). In the papers [9], [10] the algorithmic use of maximum neighborhood orderings is treated systematically.

Acknowledgments. We wish to acknowledge the anonymous referees for suggestions leading to improvements in the presentation of the results. In particular, we used the elegant proof of one of the referees for implication (iv) $\Longrightarrow$ (i) in Theorem 4. We are grateful to H.-J. Bandelt for discussions on this topic which also led to the term "dually chordal graphs."

REFERENCES

[1] R. P. Anstee and M. Farber, *Characterizations of totally balanced matrices*, J. Algorithms, 5 (1984), pp. 215–230.
[2] G. Ausiello, A. D'Atri, and M. Moscarini, *Chordality properties on graphs and minimal conceptual connections in semantic data models*, J. Comput. System Sci., 33 (1986), pp. 179–202.
[3] H.-J. Bandelt, M. Farber, and P. Hell, *Absolute reflexive retracts and absolute bipartite retracts*, Discrete Appl. Math., 44 (1993), pp. 9–20.
[4] H.-J. Bandelt and E. Prisner, *Clique graphs and Helly graphs*, J. Combin. Theory Ser. B, 51 (1991), pp. 34–45.
[5] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, *On the desirability of acyclic database schemes*, Journal ACM, 30 (1983), pp. 479–513.
[6] H. Behrendt and A. Brandstädt, *Domination and the Use of Maximum Neighborhoods*, Tech. report SM–DU–204, Department of Math, University of Duisburg, Germany, 1992.
[7] C. Berge, *Hypergraphs*, North–Holland, Amsterdam, 1989.
[8] A. Brandstädt, *Classes of bipartite graphs related to chordal graphs*, Discrete Appl. Math., 32 (1991), pp. 51–60.
[9] A. Brandstädt, V. D. Chepoi, and F. F. Dragan, *The algorithmic use of hypertree structure and maximum neighborhood orderings*, Discrete Appl. Math., to appear.
[10] A. Brandstädt, V. D. Chepoi, and F. F. Dragan, *Clique r–domination and clique r–packing problems on dually chordal graphs*, SIAM J. Discrete Math., 10 (1997), pp. 109–127.
[11] A. E. Brouwer, P. Duchet, and A. Schrijver, *Graphs whose neighborhoods have no special cycles*, Discrete Math., 47 (1983), pp. 177–182.
[12] P. Buneman, *A characterization of rigid circuit graphs*, Discrete Math., 9 (1974), pp. 205–212.
[13] R. Chandrasekaran and A. Tamir, *Polynomially bounded algorithms for locating p-centers on a tree*, Math. Programming, 22 (1982), pp. 304–315.
[14] G. J. Chang and G. L. Nemhauser, *The k-domination and k-stability problems on sun-free chordal graphs*, SIAM J. Alg. Discrete Meth., 5 (1984), pp. 332–345.
[15] G. A. Dirac, *On rigid circuit graphs*, Abh. Math. Sem. Univ. Hamburg, 25 (1961), pp. 71–76.
[16] F. F. Dragan, *Centers of Graphs and the Helly Property*, Ph.D. thesis, Department of Mathematics and Cybernetics, Moldova State University, Moldova, 1989 (in Russian).
[17] F. F. Dragan, *Conditions for coincidence of local and global minima for the eccentricity function on graphs and the Helly property*, Res. Appl. Math. and Inform. Kishinev, 1990, pp. 49–56 (in Russian).
[18] F. F. Dragan, *HT–graphs: Centers, connected r–domination and Steiner trees*, Computer Sci. J. Moldova, 1 (1993), pp. 64–83.
[19] F. F. Dragan and A. Brandstädt, *r–Dominating cliques in graphs with hypertree structure*, Discrete Math., 162 (1996), pp. 93–108.
[20] F. F. Dragan, C. F. Prisacaru and V. D. Chepoi, *r–Domination and p–center problems on graphs: Special solution methods and graphs for which this method is usable*, Preprint MoldNIINTI, N. 948–M88, Department of Mathematics and Cybernetics, Kishinev State University, Moldava, 1987 (in Russian).
[21] F. F. Dragan, C. F. Prisacaru, and V. D. Chepoi, *Location problems in graphs and the Helly property*, Diskret. Mat., 4 (1992), pp. 67–73 (in Russian).
[22] F. F. Dragan and V. I. Voloshin, *Hypertrees and associated graphs*, Tech. report DMC-MSU–0512, Department of Mathematics and Cybernetics, Moldova State University, Moldova, 1992.
[23] P. Duchet, *Propriete de Helly et problemes de representation*, Colloq. Intern. CNRS 260, Problemes Combin. et Theorie du Graphes, Orsay, France, 1976, pp. 117–118.
[24] P. Duchet, *Classical perfect graphs: An introduction with emphasis on triangulated and interval graphs*, Ann. Discrete Math., 21 (1984), pp. 67–96.
[25] R. Fagin, *Degrees of acyclicity for hypergraphs and relational database schemes*, J. ACM, 30 (1983), pp. 514–550.

[26] M. Farber, *Characterizations of strongly chordal graphs*, Discrete Math., 43 (1983), pp. 173–189.
[27] C. Flament, *Hypergraphes arbores*, Discrete Math., 21 (1978), pp. 223–227.
[28] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[29] M. C. Golumbic, *Algorithmic aspects of intersection graphs and representation hypergraphs*, Graphs Combin., 4 (1988), pp. 307–321.
[30] M. C. Golumbic and C. F. Goss, *Perfect elimination and chordal bipartite graphs*, J. Graph Theory, 2 (1978), pp. 155–163.
[31] A. J. Hoffman, A. W. J. Kolen, and M. Sakarovitch, *Totally balanced and greedy matrices*, SIAM J. Alg. Discrete Meth., 6 (1985), pp. 721–730.
[32] J. Lehel, *A characterization of totally balanced hypergraphs*, Discrete Math., 57 (1985), pp. 59–65.
[33] A. Lubiw, *Doubly lexical orderings of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.
[34] M. Moscarini, *Doubly chordal graphs, Steiner trees and connected domination*, Networks, 23 (1993), pp. 59–69.
[35] R. Paige and R. E. Tarjan, *Three partition refinement algorithms*, SIAM J. Comput., 16 (1987), pp. 973–989.
[36] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
[37] Y. Shibata, *On the tree representation of chordal graphs*, J. Graph Theory, 12 (1988), pp. 421–428.
[38] J. P. Spinrad, *Doubly lexical ordering of dense 0–1–matrices*, Inform. Process. Lett., 45 (1993), pp. 229–235.
[39] J. L. Szwarcfiter and C. F. Bornstein, *Clique graphs of chordal and path graphs*, SIAM J. Discrete Math., 7 (1994), pp. 331–336.
[40] R. E. Tarjan and M. Yannakakis, *Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.
[41] V. I. Voloshin, *Properties of triangulated graphs*, Issledovaniye operaziy i programmirovanie, Kishinev, 1982, pp. 24–32 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 456–467, August 1998

008

STRING NONINCLUSION OPTIMIZATION PROBLEMS*

ANATOLY R. RUBINOV† AND VADIM G. TIMKOVSKY‡

Abstract. For every string inclusion relation there are two optimization problems: find a longest string included in every string of a given finite language, and find a shortest string including every string of a given finite language. As an example, the two well-known pairs of problems, the longest common substring (or subsequence) problem and the shortest common superstring (or supersequence) problem, are interpretations of these two problems.

In this paper we consider a class of opposite problems connected with string noninclusion relations: find a shortest string included in no string of a given finite language and find a longest string including no string of a given finite language. The predicate "string α is not included in string β" is interpreted as either "α is not a substring of β" or "α is not a subsequence of β". The main purpose is to determine the complexity status of the string noninclusion optimization problems. Using graph approaches we present polynomial-time algorithms for the first interpretation and NP-hardness proofs for the second. We also discuss restricted versions of the problems, correlations between the string inclusion and noninclusion problems, and generalized problems which are the string inclusion problems for one language and the string noninclusion problems for another.

In applications the string inclusion problems are used to find a similarity between any structures which can be represented by strings. Respectively, the noninclusion problems can be used to find a nonsimilarity. Such problems occur in computational molecular biology, data compression, pattern recognition, and flexible manufacturing. The above generalized problems arise naturally in all of these applied areas. Apart from this practical reason, we hope that studying the string noninclusion problems will yield deeper understanding of the string inclusion problems.

Key words. string inclusion, longest, shortest, common, subsequence, substring, supersequence, superstring, polynomial-time algorithm, NP-hard problem

AMS subject classifications. 68Q20, 68Q25

PII. S0895480192234277

1. Introduction.

An *alphabet* is a nonempty finite set; a *symbol* is an element of the alphabet. Let A be an alphabet and $\mathbb{N}_m = \{1, 2, ..., m\}$. Then any mapping $\alpha : \mathbb{N}_m \to A$ is a *string* on A of *length* m and $\alpha(i)$ is the symbol at the ith *position* (the ith symbol) of α, $i \in \mathbb{N}_m$. We use the following notation: $|\alpha| = m$, $\alpha = \alpha(1)\alpha(2)...\alpha(m)$.

Let α and β be strings on A, $m = |\alpha|$, $n = |\beta|$. If there is a mapping $p : \mathbb{N}_m \to \mathbb{N}_n$ such that $p(1) < p(2) < \cdots < p(m)$ and $\alpha(i) = \beta(p(i))$ for all $i \in \mathbb{N}_m$, then α is a *subsequence* of β, β is a *supersequence* of α, and p is the *location* of α in β. In this case we write $\alpha \leq \beta$ and call $p(i)$ the ith *component* of p. Set $p(0) = 0$ and call a location p *minimal* if

$$[\, i \in \mathbb{N}_m \,\&\, p(i-1) < j < p(i)\,] \;\Rightarrow\; \beta(j) \neq \beta(p(i)),$$

i.e., p is componentwise minimal among locations of α in β. If there is a location p such that $i \in \mathbb{N}_{m-1} \Rightarrow p(i+1) = p(i) + 1$, then α is a *substring* of β, and β is a *superstring* of α. In this case we write $\alpha \trianglelefteq \beta$ and call the location p an *insertion*. If

*Received by the editors July 13, 1992; accepted for publication (in revised form) July 31, 1997; published electronically July 7, 1998.

http://www.siam.org/journals/sidma/11-3/23427.html

†National Transport Institute, Moscow, Russia (anatoly@rubinov.msk.su).

‡Computer Science Department, Economic Forecasting Institute, Russian Academy of Sciences, 32 Krasikova Street, Moscow 117418, Russia (timko@ecfor.msk.su). Current address: Star Data Systems, Inc., Commerce Court South, 30 Wellington Street S.W., Suite 300, P.O. Box 283, Toronto, Ontario M5L 1G1, Canada (vtimkovs@stardata.ca).

there is an insertion p such that $p(1) = 1$ or $p(m) = n$, then α is a *prefix* or a *suffix* of β of length m, respectively. In these cases we write $\alpha = Pref_m(\beta)$ or $\alpha = Suff_m(\beta)$.

A string $\gamma = \alpha\beta$ of length $m + n$ on A is the *concatenation* of α and β if $[i \in \mathbb{N}_m \Rightarrow \gamma(i) = \alpha(i)]$ & $[j \in \mathbb{N}_n \Rightarrow \gamma(m + j) = \beta(j)]$. Let $\mathbb{N}_0 = \emptyset$. For convenience we use the *empty string* ϵ identified by the mapping $\epsilon : \mathbb{N}_0 \to A$ with the following properties: $|\epsilon| = 0$, $\epsilon\alpha = \alpha = \alpha\epsilon$, $\epsilon \leq \alpha$, $\epsilon \trianglelefteq \alpha$, $Pref_0(\alpha) = \epsilon = Suff_0(\alpha)$ for every string α on A. Set $\alpha^0 = \epsilon$, $\alpha^1 = \alpha$, $\alpha^2 = \alpha\alpha$, the *square* of α, $\alpha^n = \alpha\alpha^{n-1}$, the nth *power* of α. Let us say that a string α has *periodicity* k, where k is natural, if $k < |\alpha|$ and there is a natural n such that $\alpha = [Pref_k(\alpha)]^n$. Call a string *periodic* if it has a periodicity and *nonperiodic* otherwise. The *period* of a periodic string is its minimal periodicity.

A *language* on an alphabet A is a nonempty set of strings on A, A^* is the language of all strings on A, $A^n = \{\alpha \in A^* : |\alpha| = n\}$, $A^{(n)} = \{\epsilon\} \cup A^1 \cup A^2 \cup \cdots \cup A^n$. Let L be a finite language on A. Then $|L|$ is the *cardinality* (the number of strings), $\|L\| = \sum_{\alpha\in L} |\alpha|$ is the *length*, $\lfloor L \rfloor = \min_{\alpha\in L}\{|\alpha|\}$ is the *thickness*, $\lceil L \rceil = \max_{\alpha\in L}\{|\alpha|\}$ is the *height* of L. An integer mapping $t : L \to \mathbb{N}_{\lceil L \rceil}$ with $\alpha \in L \Rightarrow t(\alpha) \leqslant |\alpha|$ is a *transversal* of L. Numbers $t(\alpha)$ are *components* of t. Thus, a transversal t determines the position $t(\alpha)$ in each string α of L.

2. Problem classification. For a given language one can consider *string inclusion relations* R interpreting the predicate "string α is included in string β". For example,

(seq) $\alpha R\beta \Longleftrightarrow \alpha \leq \beta$: α is included in β as a subsequence,

(str) $\alpha R\beta \Longleftrightarrow \alpha \trianglelefteq \beta$: α is included in β as a substring.

For a given string inclusion relation $R \subseteq L^2$ we define: $L_R = \{\alpha \in A^* : \beta \in L \Rightarrow \alpha R\beta\}$, $L^R = \{\alpha \in A^* : \beta \in L \Rightarrow \beta R\alpha\}$, i.e., the set of all strings on A included in (or including) every string of L, respectively. Herein we have the following two natural problems.

String inclusion optimization problems.

(Sub) Find a longest string in L_R and

(Sup) find a shortest string in L^R.

Note that $L_\leq$ and $L^\leq$ are the sets of *common subsequences* and *common supersequences*; $L_\trianglelefteq$ and $L^\trianglelefteq$ are the sets of *common substrings* and *common superstrings* for L. The string inclusion optimization problems in the interpretations (seq) and (str) are the

(LCS) *Longest common subsequence,*
(LCSS) *Longest common substring,*
(SCS) *Shortest common supersequence,*
(SCSS) *Shortest common superstring*

problems. These problems are well known and applied in molecular biology, data compression, and flexible manufacturing [9, 12, 13].

Example 2.1. For the language {413, 2343, 432} on the alphabet {1, 2, 3, 4}: LCS = 43, SCS = 234132, LCSS = 4 or 3, SCSS = 41323432 or 23432413.

Every string inclusion relation R has a complement called the *string noninclusion relation* $\not R$. For example,

(seq) $\alpha \not R \beta \Longleftrightarrow \alpha \not\leq \beta$: α is not included in β as a subsequence,

(str) $\alpha \not R \beta \Longleftrightarrow \alpha \not\trianglelefteq \beta$: α is not included in β as a substring.

Call α a *nonsubsequence* of β, β a *nonsupersequence* of α in the case (seq), and call α a *nonsubstring* of β, β a *nonsuperstring* of α in the case (str). Together with the sets L_R and L^R, we also consider the sets $L_{\not R}$ and $L^{\not R}$ of all strings on A included in (or including) no string of L, respectively. For string noninclusion relations one can consider opposite optimization problems formally exchanging the terms "longest," "shortest" and replacing R by $\not R$.

String noninclusion optimization problems.

(Sub) Find a shortest string in $L_{\not R}$ and

(Sup) find a longest string in $L^{\not R}$.

Call $L_{\not\leq}$ and $L^{\not\leq}$ the sets of *common nonsubsequences* and *common nonsupersequences*, $L_{\not\sqsubseteq}$ and $L^{\not\sqsubseteq}$ the sets of *common nonsubstrings* and *common nonsuperstrings* for L, respectively. The string noninclusion optimization problems in the interpretations (seq) and (str) are the

(SCNS) *Shortest common nonsubsequence,*
(LCNS) *Longest common nonsupersequence,*
(SCNSS) *Shortest common nonsubstring,*
(LCNSS) *Longest common nonsuperstring*

problems. It is easy to see that, in contrast to the SCNS and SCNSS problems, the LCNS and LCNSS problems can have no solution because $L^{\not\leq}$ and $L^{\not\sqsubseteq}$ may be infinite.

Example 2.2. For the language {111, 222, 1212, 2211} on the alphabet {1, 2}, SCNS = LCNS = 1122 or 1221 or 2112 or 2121, SCNSS = 112 or 122, and there is no LCNSS because the string $(121)^k$ is a common nonsuperstring for every natural k. However, for the language {11, 122, 21, 22} on the same alphabet, LCNS = LCNSS = 12.

The paper is devoted to these four problems. We assume that the language L does not contain the empty string, is not empty, and is "inclusion free," i.e.,

$$(\mathrm{F}) \qquad [\alpha, \beta \in L \ \&\ \alpha \text{ is included in } \beta\,] \implies \alpha = \beta.$$

If (F) is false then we can delete α or β from L in the case (Sub) or (Sup), respectively. Besides, for the case (Sup) we assume that the language L is "alphabetwise closed," i.e.,

$$(\mathrm{C}) \qquad \forall a \in A \quad \exists n(a) \geqslant 2 : \quad a^{n(a)} \in L.$$

If (C)[1] is false, then (Sup) problems have no solution in the case $a^n \notin L$ for all natural n, or all strings containing a can be deleted from L, and a can be deleted from A in the case $a \in L$. The assumption (C) is not only necessary, but it is also sufficient for the existence of an LCNS, because the length of any common nonsupersequence does not exceed $\sum_{a \in A}[n(a) - 1]$. However, it is not sufficient for the existence of an LCNSS, as shown in Example 2.2. Thus, the LCNSS search problem has a sense only for languages, for which the following question has a positive answer.

LCNSS existence problem. Does there exist an LCNSS for L?

Below we consider the LCNSS problem as the union of the existence and search problems.

[1] We use the same notation for symbols and one-symbol strings.

String noninclusion optimization problems were apparently introduced in [14]. The same paper proposes the conjecture: if an LCNSS exists, then LCNSS length is bounded by the language length.

In this paper we prove this conjecture and determine the complexity status of string noninclusion optimization problems. We suggest polynomial-time algorithms in the interpretation (str) and prove NP-hardness in the interpretation (seq). We also show that the SCNS and LCNS problems in the case of bounded language cardinality are solvable in polynomial time. Since the main purpose of this paper is to determine the complexity status of the above problems, we present simple polynomial-time algorithms, but not efficient ones. Related issues and applications are discussed as well.

3. Longest common nonsuperstring problem. For a string set V without the empty string on an alphabet A construct the directed graph G_V with vertex set V and arc set E determined by the rule:

$$(\alpha, \beta) \in E \iff Suff_{|\alpha|-1}(\alpha) = Pref_{|\alpha|-1}(\beta).$$

The arc (α, β) arises when $|\alpha| \leqslant |\beta|+1$ and there is a string in A^* of length $|\beta|+1$ with prefix α and suffix β. This string is denoted as $[\alpha, \beta]$. Note that it is not necessarily in V. In particular, if $|\alpha| = 1$, then there are arcs from α to all other vertices of G_V; $Suff_{|\alpha|-1}(\alpha) = \beta \Rightarrow (\alpha, \beta) \in E$.

Example 3.1. $G_{English}$ has the arcs $(word, order)$, $(a, part)$, $(there, here)$; herein $[word, order] = worder$, $[a, part] = apart$, $[there, here] = there$.

For every route[2] $M = (\sigma_1, \sigma_2, ..., \sigma_k)$ in G_V define the string

$$f(M) = \sigma_1(1)\sigma_2(1)...\sigma_{k-1}(1)\sigma_k$$

contained in $\{\sigma_1, \sigma_2, ..., \sigma_k\}^{\lhd}$. Informally speaking a string on A belongs to the image of the mapping f if it can be "paved" by strings of V. Among insertions of σ_l in $f(M)$, where $l \in N_k$, we will distinguish the *proper insertion* $p_l(1) = l$, i.e., the insertion starting with position l of $f(M)$.

Remark 3.1. Note that G_{A^n} is the well-known graph related to de Bruijn's sequence [4], and the mapping f is a one-to-one correspondence between routes of length k in G_{A^n} and strings of length $n + k - 1$ on the alphabet A.

Now let S be the set of proper suffixes of L, i.e., all strings written as $Suff_n(\alpha)$, where $\alpha \in L$, $0 < n < |\alpha|$. For every nonempty string ω on A define the route in G_S

$$g(\omega) = (\sigma_1, \sigma_2, ..., \sigma_{|\omega|}),$$

where σ_i is the longest string of S included in ω as a substring starting from the ith position. Since (C) is true, $a \in A \Rightarrow a \in S$ and so the choice of σ_i is always possible. It is important to observe that the inequality $|\sigma_i| \leqslant |\sigma_{i+1}| + 1$ and the equality $Suff_{|\sigma_i|-1}(\sigma_i) = Pref_{|\sigma_i|-1}(\sigma_{i+1})$ follow from the fact that σ_i and σ_{i+1} are included in ω as substrings starting from the ith and $(i + 1)$th positions, respectively, and the longest length requirement. Thus, the arc (σ_i, σ_{i+1}) exists in fact, i.e., the definition of the route $g(\omega)$ is correct. It is easy to see that $f(g(\omega)) = \omega$.

Let Γ_S be a subgraph of G_S with vertex set S and arcs (α, β) with the property: α is the longest prefix of $[\alpha, \beta]$ contained in $L \cup S$, i.e., among suffixes of L there are no prefixes of $[\alpha, \beta]$ longer than α.

[2]Unlike a path, a route can intersect itself.

Example 3.2. If $L = English$, then the arc $(ove, venir)$ of G_S is not in Γ_S since $ove = Suff_3(love)$, $venir = Suff_5(souvenir)$, $[ove, venir] = ovenir$, but $oven = Pref_4(ovenir) \in L$. Neither is the arc (e, t) of G_S in Γ_S since $e = Suff_1(love)$, $t = Suff_1(let)$, $[e, t] = et = Pref_2(et) = Suff_2(net) \in S$. However, the arc (ee, ea) is in Γ_S since $ee = Suff_2(tree)$, $ea = Suff_2(tea)$, $[ee, ea] = eea$, where $eea \notin L \cup S \ni ee$.

Below we show that Γ_S is constructed so that there is a correspondence between the set of routes in Γ_S and the set of nonsuperstrings of L.

LEMMA 3.1. (a) *If M is a route in Γ_S, then $f(M) \in L^{\not\exists}$;* (b) *if $\omega \in L^{\not\exists}$, then $g(\omega)$ is a route in Γ_S.*

Proof. Let (a) be false and $M = (\sigma_1, \sigma_2, ..., \sigma_k)$ be a route in Γ_S such that $\sigma \in L$ and $\sigma \trianglelefteq \varphi = f(M)$. Then we will find an arc in M, which cannot be an arc of Γ_S.

Let p be an insertion of σ in φ, and let p_l be the proper insertion of σ_l in φ, where $l \in \mathbb{N}_k$. Then $p_1(|\sigma_1|) \leqslant p_2(|\sigma_2|) \leqslant \cdots \leqslant p_k(|\sigma_k|) = |\varphi|$. This chain of inequalities follows immediately from the definition of arcs of the graph G_S.

Choose minimal natural j in $\mathbb{N}_k$ such that $p(|\sigma|) \leqslant p_j(|\sigma_j|)$ and show that Γ_S does not contain the arc (σ_{j-1}, σ_j). For this purpose we will find a string $\pi \in L \cup S$, which is a prefix of $[\sigma_{j-1}, \sigma_j]$ with $|\pi| > |\sigma_{j-1}|$. If $p(1) \geqslant p_j(1)$, then $\sigma \trianglelefteq \sigma_j$, which contradicts (F). If $p(1) < p_j(1)$, then $p(1) \leqslant p_{j-1}(1)$ and there is the chain of inequalities:

$$p(1) \leqslant p_{j-1}(1) \leqslant p_{j-1}(|\sigma_{j-1}|) < p(|\sigma|) \leqslant p_j(|\sigma_j|).$$

This means that $[\sigma_{j-1}, \sigma_j] = \varphi(p_{j-1}(1)...\varphi(p_j(|\sigma_j|))$ has the prefix

$$\pi = \varphi(p_{j-1}(1))...\varphi(p(|\sigma|)),$$

a suffix of σ, which is longer than σ_{j-1}. But this contradicts the existence of the arc (σ_{j-1}, σ_j) in Γ_S.

Now let (b) be false, and for a string $\omega \in L^{\not\exists}$ the route $g(\omega) = (\sigma_1, \sigma_2, ..., \sigma_{|\omega|})$ in G_S is not a route in Γ_S. Then for some natural $i \in \mathbb{N}_{|\omega|-1}$ the arc (σ_i, σ_{i+1}) is not in Γ_S. This means that $[\sigma_i, \sigma_{i+1}]$ has a prefix $\pi \in L \cup S$ with $|\pi| > |\sigma_i|$. If $\pi \in S$, then it contradicts the choice rule of σ_i in $g(\omega)$. If $\pi \in L$, then the inclusions $\pi \trianglelefteq [\sigma_i, \sigma_{i+1}] \trianglelefteq \omega$ contradict $\omega \in L^{\not\exists}$. □

Remark 3.2. Note that the last vertex of the route $g(\omega)$ is a one-symol suffix. It is easy to show that the mappings f and g determine a one-to-one correspondence between $L^{\not\exists}$ and the set of routes in Γ_S ending in one-symol suffixes. Besides, an arbitrary route $M = (\sigma_1, \sigma_2, ..., \sigma_k)$ in Γ_S can be extended to the route $M' = (\sigma_1, \sigma_2, ..., \sigma_k, Suff_{|\sigma_k|-1}(\sigma_k), Suff_{|\sigma_k|-2}(\sigma_k), ..., Suff_1(\sigma_k))$.

Lemma 3.1 reduces a consideration of the LCNSS problem to the analysis of the graph Γ_S. Bounded length of common nonsuperstrings for L means bounded length of routes in Γ_S, i.e., Γ_S is acyclic. Thus, the following theorem is proved.

THEOREM 3.1. *If the graph Γ_S is acyclic, then M is the longest path in it iff $f(M)$ is an LCNSS for the language L. If $M = (\sigma_1, \sigma_2, ..., \sigma_k)$ is a closed route in Γ_S, i.e., $\sigma_1 = \sigma_k$, then for any natural n the string $[Pref_{k-1}(f(M))]^n$ is a common nonsuperstring for L and so there is no LCNSS for L.*

COROLLARY 3.1. *The LCNSS problem is solvable in polynomial time.*

Proof. The number of proper suffixes of L, and so the construction time of Γ_S, are bounded by a polynomial in $\|L\|$. Besides, the graph cycle existence and the acyclic graph longest path problems are solvable in polynomial time [8]. □

COROLLARY 3.2. *If an LCNSS exists for L, then $|LCNSS| \leqslant |S|$.*

This easy corollary from Theorem 3.1 proves the conjecture from [14], namely, if an LCNSS exists for L, then $|\text{LCNSS}| \leqslant \|L\|$. It is true because $|S| \leqslant \|L\| - |L|$. The estimate is exact as shown in the following example.

Example 3.3. For the language $L = \{11, 22, 12\}$ on the alphabet $\{1, 2\}$ we have: $|\text{LCNSS}| = |S| = 2$, $\|L\| = 6$, $|L| = 3$, LCNSS=21.

The length of L is the natural size of string noninclusion problems. That is why we use it to estimate problem complexities. On the other hand, as the following result shows, if an LCNSS exists for L, then the cardinality and length of the language L depend exponentially on its thickness.

THEOREM 3.2. *If an LCNSS exists for L, then*

$$|L| \geqslant \frac{|A|^{\lfloor L \rfloor}}{\lfloor L \rfloor} \quad \textit{and} \quad \|L\| \geqslant |A|^{\lfloor L \rfloor}.$$

Proof. Let $M = (\sigma_1, \sigma_2, ..., \sigma_k)$ be a closed route in the graph G_{A^r}, where $r = \lfloor L \rfloor$. As the length of common nonsuperstrings is bounded, there is a natural n such that $[Pref_{k-1}(f(M))]^n$ is the superstring of a string $\omega_M \in L$. If another closed route M' in the same graph has no common vertices with M, then the string ω_M and the corresponding string $\omega_{M'}$ have no common substring of length r and so $\omega_M \neq \omega_{M'}$. Thus to prove the first inequality it is sufficient to find at least $\frac{|A|^r}{r}$ pairwise nonintersecting closed routes in G_{A^r}. Define a function $h : A^r \to A^r$ taking $h(\alpha) = \alpha(2)\alpha(3)...\alpha(r)\alpha(1)$. Set $h^0(\alpha) = \alpha$ and $h^n(\alpha) = h(h^{n-1}(\alpha))$ where $n \geqslant 1$. Obviously, $M_\alpha = (\alpha, h(\alpha), h^2(\alpha), ..., h^r(\alpha))$ is a closed route in G_{A^r}.

Let q be the period of α. Then M_α comprises $\frac{r}{q}$ rounds of a cycle on q vertices. Thus, vertex set A^r of G_{A^r} is covered by pairwise nonintersecting cycles M_α, $\alpha \in A^r$, and each cycle has at most r vertices. So the number of these cycles is at least $\frac{|A|^r}{r}$. The second inequality follows from the first one because $|L|\lfloor L \rfloor \leqslant \|L\|$. □

Remark 3.3. Considering the cycles M_α in G_{A^r} we can suggest a trivial combinatorial proof of Fermat's (little) theorem [2]: for any natural m, prime r divides $m^r - m$. Take an alphabet A with $|A| = m$. Since r is prime, the only periodicity of a periodic string of length r is one. So with the exception of strings a^r, where $a \in A$, the other $m^r - m$ strings of A^r are distributed among r-vertex cycles M_α, i.e., $m^r - m \equiv 0 (\text{mod } r)$.

Similar reasoning leads to the generalization of Fermat's theorem to the case of composite r, which was formerly unknown to the authors. Let $r = p_1^{n_1} p_2^{n_2} ... \, p_l^{n_l}$ be the decomposition of r into the product of prime powers, and let q be a divisor of r, where $q \neq r$. Then all m^q strings of length q raised to power of $\frac{r}{q}$ form the set of all periodic strings of length r with periodicity q (this set also contains all periodic strings with periodicity q', where q' is a divisor of q). Implementing the inclusion-exclusion principle, it is not difficult to show that the number of all nonperiodic strings of length r is

$$m(r) \quad = \sum_{k_1, k_2, ..., k_l \in \{0,1\}} (-1)^{k_1 + k_2 + \cdots + k_l} \, m^{p_1^{n_1 - k_1} p_2^{n_2 - k_2} ... p_l^{n_l - k_l}}.$$

So $m(r) \equiv 0 (\text{mod } r)$. In the case $l = 1$ this proposition coincides with Euler's theorem.

4. Shortest common nonsubstring problem. Polynomial solvability of the SCNSS (and LCSS) problem trivially follows from the fact that the total length of all substrings of strings in L does not exceed $\|L\|^3$. Considering these substrings in a list

ordered by length and lexicographically ordered among pieces of the same length, we can easily find an SCNSS as the first lexicographic hole.[3]

5. Longest common nonsupersequence problem. As we have shown in the second section, the LCNS existence problem reduces to checking the condition (C) and so it is trivial. However, the LCNS search problem is essentially more difficult. To show this, we need the following well-known NP-complete problem [3].

Independent set problem. Given an undirected graph G with vertex set V, edge set E, and a natural $k \leqslant |V|$, does G have an independent set[4] $I \subseteq V$ of at least k vertices?

THEOREM 5.1. *The LCNS search problem is NP-hard.*

Proof. Let us show a reduction from the independent set problem to the LCNS decision problem: given a finite language L on an alphabet A and a natural l, does there exist a common nonsupersequence ω for L of length at least l? Set

$$A = V, \quad l = k, \quad L = \{vv : v \in V\} \cup \{uv, vu : \{u, v\} \in E\}.$$

The symbol set of any common nonsupersequence for L is an independent set in G, and the vertices of any independent set in G written in arbitrary order make a common nonsupersequence for L. So $I = \{v_1, v_2, ..., v_k\} \Leftrightarrow \omega = v_1 v_2 ... v_k$. □

Example 5.1. If V = {1,2,3,4,5}, E = {{1,2},{2,3},{2,5},{3,4},{3,5}}, $k = 3$, I = {1,4,5}, then A = {1,2,3,4,5}, L = {11,22,33,44,55} ∪ {12,21,23,32,25,52,34,43,35,53}, $l = 3$, $\omega = 145$.

Remark 5.1. From Theorem 5.1 proof follows that the LCNS search problem remains NP-hard even if every string of L has length two. Besides, NP-hardness proofs of the restricted version of the LCNS search problem with bounded alphabet size have been proposed by Jiang [6] for the case $|A| = 3$ and by Zhang [15] for the case $|A| = 2$. These nontrivial proofs also employ the reduction from the independent set problem.

Now we show that LCNS can be found in polynomial time if the language cardinality is bounded. Let $\#A = \{\#\} \cup A$, where the symbol # does not belong to A, let $\#L = \{\#\alpha : \alpha \in L\}$, and let V be the set of all transversals of $\#L$. Define a directed graph G with the vertex set V and arcs labelled by symbols of A. The arc (u, v) labelled by $A(u, v)$ exists iff $u \neq v$ and $u(\alpha) + 1 = v(\alpha)$ for strings $\alpha \in \#L$ containing $A(u, v)$ in the $[u(\alpha) + 1]$th position and $u(\alpha) = v(\alpha)$ for other strings $\alpha \in \#L$.

Example 5.2. If $A = \{a, b\}$, $\#L = \{\#aa, \#ab, \#bb\}$, then $u = \langle 1, 1, 1\rangle$,[5] $v = \langle 2, 2, 1\rangle, w = \langle 2, 3, 2\rangle \in V$; $(u, v), (v, w)$ are arcs labelled by $A(u, v) = a$, $A(v, w) = b$; $(u, w), (v, u)$ are not arcs.

It is easy to see that the graph G is acyclic. The vertex $v_0 \in V$ is called a *source* iff $\forall \omega \in \#L : v_0(\omega) = 1$, and every vertex $v \in V$ is called *final* iff $\exists \omega \in \#L : v(\omega) = |\omega|$ and *nonfinal* otherwise. Let $\mathcal{P}$ be the set of all paths in G starting from the source and having at least one arc, and let $\mathcal{P}'$ be the subset of $\mathcal{P}$ consisting of paths containing no final vertices. Define a mapping $f : \mathcal{P} \to A^*$ taking the path $P = (v_0, v_1, v_2, ..., v_{k-1}, v_k) \in \mathcal{P}$ to the string $f(P)$ by $f(P) = A(v_0, v_1)A(v_1, v_2)...A(v_{k-1}, v_k)$.

LEMMA 5.1. *f is a one-to-one correspondence between $\mathcal{P}'$ and $L^{\not\leq}$.*

[3] The first string of A^* that is not in the list.

[4] A set of pairwise nonadjacent vertices.

[5] We write $t = \langle t(\alpha_1), ..., t(\alpha_n)\rangle$ if t is a transversal of the language $\{\alpha_1, ..., \alpha_n\}$.

Proof. Since labels of arcs going from the same vertex are different, the mapping f is injective. Besides, $L^{\not\subseteq} \subseteq \operatorname{Im} f$ due to the assumption (C): there is an arc labelled by a that goes from a vertex v if $v(\#a^{n(a)}) \leqslant n(a)$, i.e., $\operatorname{Im} f$ contains all strings including each symbol a from A at most $n(a) - 1$ times.

Now let $P = (v_0, v_1, ..., v_k) \in \mathcal{P}$, $\alpha \in L$, $t_j = v_j(\#\alpha) - 1$, $j = 0, 1, ..., k$, and $\pi = Pref_{t_k}(\alpha)$. To prove the lemma it is sufficient to show that π is the longest prefix of α included in $f(P)$ as a subsequence. Note that $0 \leqslant t_j - t_{j-1} \leqslant 1$, where $j \in \mathbb{N}_k$, so one can define a mapping $p : \mathbb{N}_{t_k} \to \mathbb{N}_k$ by the formula

$$p(i) = \min_{j \in \mathbb{N}_k \,\&\, i = t_j} \{j\}, \qquad i \in \mathbb{N}_{t_k}.$$

It is easy to see that p determines a location of π in $f(P)$. Besides, this location is minimal and $\alpha(t_k + 1) \neq f(P)(j)$ if $j > p(t_k)$. □

Lemma 5.1 reduces the LCNS search problem to the search for a longest path in $\mathcal{P}'$. Note that G has no cycles and arcs from final vertices to nonfinal vertices. So contracting all final vertices of G to a single *terminal* t and removing any loops that arise, we obtain an acyclic graph Γ.

THEOREM 5.2. *$(v_0, v_1, ..., v_k, t)$ is a longest path in Γ from v_0 to t iff $f(v_0, v_1, ..., v_k)$ is an LCNS for L.*

COROLLARY 5.1. *If the cardinality of L is bounded, then an LCNS for L can be found in polynomial time.*

Proof. The number of transversals of the language $\#L$, and so the number of vertices in G, do not exceed $\|L\|^{|L|}$ and there is a polynomial-time longest path algorithm for acyclic directed graphs [8]. □

6. Shortest common nonsubsequence problem. To show the intractability of the SCNS problem we need the following well-known NP-complete problem [3].

Vertex cover problem. Given an undirected graph G with vertex set V, edge set E, and a natural $k \leqslant |V|$, does G have a vertex cover[6] $C \subseteq V$ of at most k vertices?

THEOREM 6.1. *The SCNS problem is NP-hard.*

Proof. Let us show a reduction from the vertex cover problem to the SCNS decision problem: given a finite language L on an alphabet A and a natural l, does there exist a common nonsubsequence σ for L of length at most l? Set

$$A = V, \quad l = k, \quad L = \{\, \zeta_e^{|V|} : e \in E \,\},$$

where ζ_e is any fixed string of length $|V| - 2$ including all symbols of V except the ends of the edge e. Then the symbol set of any common nonsubsequence for L of length at most l is a vertex cover in G, and the vertices of any vertex cover in G written in arbitrary order make a common nonsubsequence for L. So $C = \{v_1, v_2, ..., v_k\} \Leftrightarrow \sigma = v_1 v_2 ... v_k$. □

Example 6.1. If $V = \{1, 2, 3, 4, 5\}$, $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$, $k = 3$, $C = \{2, 3, 4\}$, then $A = \{1, 2, 3, 4, 5\}$, $L = \{\zeta^5_{\{i,i+1\}} : i \in \mathbb{N}_4\}$, $\zeta_{\{1,2\}} = 345$, $\zeta_{\{2,3\}} = 145$, $\zeta_{\{3,4\}} = 125$, $\zeta_{\{4,5\}} = 123$, $l = 3$, $\sigma = 234$.

Remark 6.1. In contrast to the LCNS search problem (see Remark 5.1) the SCNS problem can be solved in polynomial time if strings of L are of bounded length. In this case the list of all subsequences of strings of L is also bounded, so an SCNS can be found in the same way as an SCNSS (see section 4). Employing the reduction from the vertex cover set problem as well, nontrivial NP-hardness proofs of the restricted

[6] A set of vertices such that every edge is incident with at least one of the vertices.

version of the SCNS problem with bounded alphabet size have been proposed by Jiang [6] for the case $|A| = 4$ and by Middendorf [10] for the case $|A| = 2$.

Now we show that an SCNS can be found in polynomial time if the language cardinality is bounded. Again let $\#A = \{\#\} \cup A$, where $\# \notin A$, and $\#LA = \{\, \#\alpha\lambda^{\lceil L\rceil+1} : \alpha \in L \,\}$, where λ is a fixed string of length $|A|$ including all symbols of A. A position i in the string $\#\alpha\lambda^{\lceil L\rceil+1}$ of $\#LA$ is called *external* if $i > |\alpha| + 1$ and *internal* otherwise. A transversal t of a language K on A is called *uniform* and labeled by $A(t)$ if there exists a symbol $A(t)$ in A such that $\alpha \in K \Rightarrow \alpha(t(\alpha)) = A(t)$.

Let V be a set of all uniform transversals of $\#LA$. Define a directed graph G with vertex set V and arcs (u, v) with

$$\forall\omega \in \#LA: \quad u(\omega) < v(\omega)\,\&\,[\,u(\omega) < j < v(\omega) \implies \omega(j) \neq A(v)\,],$$

i.e., v is the componentwise minimal transversal among all uniform ones which are componentwise larger than u. Let the arc (u, v) have the label $A(v)$.

Example 6.2. If $A = \{a, b\}$, $L = \{aba, bba\}$, $\lambda = ab$, then $\lceil L\rceil = 3$; $\#LA = \{\#abaababababab, \#bbaababababab\}$; $u = \langle 1, 1\rangle$, $v = \langle 2, 4\rangle$, $w = \langle 3, 2\rangle$, $x = \langle 4, 4\rangle \in V$; $A(u) = \#$, $A(v) = A(x) = a$, $A(w) = b$; (u, v), (u, w), (w, x) are arcs; (u, x), (v, w), (v, x) are not arcs.

We can see that the graph G is acyclic. The vertex $v_0 \in V$ with $v_0(\omega) = 1 \quad \forall\omega \in \#LA$ is called the *source* in G. A vertex $v \in V$ is called *external* if $\forall\omega \in \#LA$ the component $v(\omega)$ is an external position of ω and *internal* otherwise.

Let $\mathcal{P}$ be the set of all paths in G starting from the source and having at least one arc, and let $\mathcal{P}'$ be the subset of paths containing at least one external vertex and at most $\lceil L\rceil + 1$ arcs. Define a mapping $f : \mathcal{P} \to A^*$ taking the path $P = (v_0, v_1, v_2, ..., v_k) \in \mathcal{P}$ to the string $f(P)$ by $f(P) = A(v_1)A(v_2)\,...\,A(v_k)$.

LEMMA 6.1. *f is a one-to-one correspondence between $\mathcal{P}'$ and $L_{\not\leq} \cap A^{(\lceil L\rceil+1)}$.*

Proof. Since labels of arcs going from the same vertex are different, the mapping f is injective. Besides, $A^{(\lceil L\rceil+1)} \subseteq \operatorname{Im} f$ due to $\lambda^{\lceil L\rceil+1} \in \#LA_{\leq}$. Now let $P = (v_0, v_1, ..., v_k) \in \mathcal{P}$, $\omega \in \#LA$. Define $p : \mathbb{N}_k \to \mathbb{N}_{|\omega|}$ setting $p(i) = v_i(\omega)$, $i \in \mathbb{N}_k$. This mapping determines the minimal location of $f(P)$ in ω, so $f(P) \in L_{\not\leq}$ iff v_k is an external vertex. □

Since the length of any SCNS does not exceed $\lceil L\rceil + 1$, Lemma 6.1 reduces the SCNS search problem to the search for a shortest path in $\mathcal{P}'$. G has no cycles and arcs going from external vertices to internal ones. So contracting all external vertices to a single *terminal* t and removing any loops that arise, we obtain an acyclic graph Γ.

THEOREM 6.2. *P is a shortest path in Γ from v_0 to t iff $f(P)$ is an SCNS for L.*

COROLLARY 6.1. *If the cardinality of L is bounded, then an SCNS for L can be found in polynomial time.*

Proof. The number of all transversals of $\#LA$, and so the number of vertices of G do not exceed $\|L\|\,(\,2\|L\| + 1\,)^{|L|}$. Besides, the shortest path problem is polynomial [8]. □

Remark 6.2. Unlike the other string inclusion and noninclusion problems considered above, the SCNS problem remains nontrivial if $|L| = 1$.

(SNS) Shortest nonsubsequence problem. Let a string σ contain all symbols of an alphabet A. Find a shortest nonsubsequence $\eta \in A^*$ of σ.

We suggest a simple SNS algorithm without using shortest path procedure. Let $\sigma_1 = \sigma$ and find the shortest prefix π_1 of σ_1 containing all symbols of A. Then

$\sigma_1 = \pi_1\sigma_2$. Repeat this operation with the string σ_2, etc. After k iterations of this operation we get $\sigma_1 = \pi_1\pi_2 ... \pi_k\sigma_{k+1}$, where σ_{k+1} does not contain all symbols of A (it may be that $|\sigma_{k+1}| = 0$). Set $\eta = \pi_1(|\pi_1|)\pi_2(|\pi_2|) ... \pi_k(|\pi_k|)a$, where $a \in A\backslash \mathrm{Im}\ \sigma_{k+1}$. In other words, the string η consists of the last symbols of the prefixes obtained and one more symbol which is not contained in σ_{k+1}.

Let us show that the string η is an SNS of σ. The rule of choosing the prefix π_i implies that the symbol $\pi_i(|\pi_i|)$ appears in it only once in the last position. So the mapping $p : \mathbb{N}_k \to \mathbb{N}_{|\sigma|}$, where $p(i) = |\pi_1| + |\pi_2| + \cdots + |\pi_i|$, determines the minimal location of $Pref_k(\eta)$ in σ, and there is no symbol a in σ_{k+1}. Thus, η is a nonsubsequence of length $k + 1$. Besides, there are no nonsubsequences shorter than η because any string of length k is a subsequence of $\pi_1\pi_2 ... \pi_k$.

7. Conclusion. The complexity status of the corresponding string inclusion and noninclusion problems is different only for the SCSS/LCNSS pair: the SCSS problem is NP-hard [3], while the LCNSS problem is solvable in polynomial time. This means that the string noninclusion problems studied here are more tractable and more "regular" than the corresponding string inclusion problems, because their complexity status is determined by an interpretation of the inclusion relation: (str) leads to polynomial-time solvability and (seq) leads to NP-hardness.

On the other hand, the corresponding string inclusion and string noninclusion problems can be solved by similar approaches. For example, the SCNS problem is reducible to the problem of finding a shortest path in the directed graph from the source to external vertices. Now call a transversal t an internal vertex if $\alpha \in \#LA \Rightarrow t(\alpha)$ has only internal positions. Then the LCS problem is reducible to finding a longest path containing internal vertices only. The LCNS algorithm described above can be similarly transformed to an SCS algorithm by modifying the definition of the terminal and by interchanging longest and shortest path algorithms.

Note that the corresponding LCSS and SCNSS, SCSS and LCNSS problems can also be solved by similar approaches. The LCSS and SCNSS problems are solved by the list of substrings of L, and to solve the SCSS problem, we can avoid a transformation of the graph G_S to Γ_S as in the LCNSS case, and instead of it, add in G_S the vertices corresponding to all strings of L and reduce the SCSS problem to the search for a shortest path containing all added vertices. This problem, however, is already NP-hard [3].

Thus we can speak about some duality between the string inclusion and noninclusion problems. It is interesting to investigate correlations between them because in practice there are problems which occupy an intermediate place between the string inclusion and noninclusion problems. An obvious example here is the *shortest consistent superstring problem* arisen from data compression practice and DNA sequencing procedures [7, 9, 12]. It involves, for two given languages of *positive* and *negative* strings, finding the shortest possible string σ such that every positive string is a substring of σ and no negative string is a substring of σ. Similar problems are found in flexible manufacturing, where the alphabet and the language determine the sets of technological operations and technology types fulfilled by a manufacturing system. The inclusion relation means the possibility to fulfill one technology within another one, and negative strings determine technological restrictions. The length of an SCNS or an SCNSS measures manufacturing system flexibility in this case since any shorter technology is fulfilled by the system [13, 14].

The shortest consistent superstring problem is one among many problems (with two languages of positive and negative strings) which can be formally generated from it

by varying the inclusion relations $\trianglelefteq$ and $\leq$, the specifications "sub" and "super" (they may be different for positive and negative strings), and the criteria "shortest" and "longest." For example, the *shortest distinguishing string-language problem* formulated by Middendorf [10] consists of finding a shortest string that is a subsequence of a single positive string and a common nonsubsequence for a language of negative strings. The NP-hardness of this problem has been proved by a reduction from the SCNS problem [10]. However, the case with a single negative string, the *shortest distinguishing string-string problem*, is solvable in polynomial time [5].

In practice there are string inclusion and noninclusion problems with a more complex interpretation of the inclusion relation. For example,

$$\alpha R \beta \iff \alpha = \gamma_1\gamma_2...\gamma_n \ \& \ \beta = \pi_0\gamma_1\pi_1\gamma_2\pi_2...\gamma_n\pi_n,$$

where $\pi_0, \pi_i, \gamma_i \in A^*$ for all $i \in N_n$, $1 \leq n \leq k$ and k is fixed. If $k = 1$, then $R = \trianglelefteq$; if $k < \infty$, then $R = \leq$. If $k = 2$, then α is included in β as a substring or as two nonoverlapping substrings.

We suppose that studies of generalizations of the string noninclusion problems in the case of an infinite language can produce interesting results related to *avoidable patterns in infinite sequences*. Consider, for example, the LCNSS problem with the language of squares $\{\sigma\sigma : \sigma \in A^*\}$. Let $A = \{1, 2, ..., n\}$. In this case it is easy to test, if $n = 1$, then LCNSS=1, if $n = 2$, then LCNSS=121 or 212, because there are just seven common nonsuperstrings: ϵ, 1, 2, 12, 21, 121, 212. If $n = 3$, then there is no LCNSS, because for any natural k the prefix $Pref_k(\tau)$, where τ is the infinite *Thue's sequence* [11] avoiding squares, may be taken to be a common nonsuperstring. For similar results, see [1, 16].

Acknowledgments. We thank Robert Irving, Tao Jiang, Pavel Pevzner, and Louxin Zhang for useful discussions. We also thank Tao Jiang and the referees for careful considerations and comments.

REFERENCES

[1] R. Bean, A. Ehrenfeucht, and G. F. McNulty, *Avoidable patterns in strings of symbols*, Pacific J. Math., 85 (1979), pp. 261–294.
[2] H. M. Edwards, *Fermat's Last Theorem*, Springer-Verlag, New York, 1977.
[3] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman, San Francisco, CA, 1979.
[4] M. Hall, Jr., *Combinatorial Theory*, Blaisdel, Waltham, MA, 1967.
[5] J.-J. Hebrard, *An algorithm for distinguishing efficiently bit-strings by their subsequences*, Theoret. Comput. Sci., 82 (1991), pp. 35–49.
[6] T. Jiang, *Private communication*, 1993.
[7] T. Jiang and M. Li, *Approximating shortest superstrings with constraints*, Theoret. Comput. Sci., 134 (1994), pp. 473–491.
[8] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.
[9] A. Lesk, ed., *Computational Molecular Biology, Sources and Methods for Sequence Analysis*, Oxford University Press, Oxford, 1988.
[10] M. Middendorf, *The shortest common non-subsequence problem is NP-complete*, Theoret. Comput. Sci., 108 (1993), pp. 365–369.
[11] A. Salomaa, *Jewels of Formal Language Theory*, Computer Science Press, Rockville, MD, 1981.
[12] J. Storer, *Data Compression: Methods and Theory*, Computer Science Press, Rockville, MD, 1988.
[13] V. G. Timkovsky, *Discrete Mathematics in Engineering*, Nauka, Moscow, 1992 (in Russian).

[14] V. G. Timkovsky, *Complexity of common subsequence and supersequence problems and related problems*, Kibernetika, 5 (1989), pp. 1–13 (in Russian). English transl. in Cybernetics, 25 (1990), pp. 565–580.

[15] L. Zhang, *On the approximation of longest common nonsupersequences and the shortest common nonsubsequences*, Theoret. Comput. Sci., 143 (1995), pp. 353–362.

[16] A. I. Zimin, *Blocking sets of terms*, Math. Sbornik, 119 (1982), pp. 363–375 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 468–486, August 1998

009

PARTIAL INTERSECTION THEOREM AND FLOWS IN ABSTRACT NETWORKS*

MARTIN KOCHOL†

Abstract. The aim of this paper is to introduce a general framework for various results regarding constructions of matroids and (generalized) polymatroids—for instance, the basic operations on (generalized) polymatroids and constructions of transversal matroids, gammoids, and their generalizations. All of them are covered by the following theorem: If $\mathbb{P}_1$ and $\mathbb{P}_2$ are generalized polymatroids in $\mathbb{R}^n \oplus \mathbb{R}^m$ and $\mathbb{R}^m$, respectively, and $\mathbb{P}'_1$ is the set of the vectors from $\mathbb{P}_1$ whose projections to $\mathbb{R}^m$ are in $\mathbb{P}_2$, then the projection of $\mathbb{P}'_1$ to $\mathbb{R}^n$ is a generalized polymatroid. An equivalent statement is obtained using a flow model that has many common features with the concept of group-valued flows.

Key words. partial intersection theorem, abstract network, (generalized, quasi) polymatroid, matroid, v-transversal, v-gammoid, operations on matroids and (generalized) polymatroids, edge-cut

AMS subject classifications. 52B40, 05B35, 90B10

PII. S0895480195295987

1. Introduction.

The Edmonds intersection theorem [E] gives a necessary and sufficient condition for two matroids to have a common independent set of cardinality at least d. A more advanced form of this theorem says that the linear system describing the intersection polytope of two polymatroids is totally dual integral. This result generalizes plenty of classical min-max relations from combinatorics (for instance, the Hall, Dilworth, and max-flow min-cut theorems) and is equivalent to many sophisticated theorems from combinatorial optimization (see, e.g., the survey article of Schrijver [S3]). Various generalizations of this theorem are in [Fr], [KC], [M1], [M2], [M3], [N3], [S3], [S4], [T], and [Wo].

A few similar results were obtained by Davies and McDiarmid [DMcD] who give a necessary and sufficient condition for two strongly base orderable matroids, (which form a proper subclass of matroids), to have k disjoint common independent sets of cardinality at least d. This theorem can be used as a general framework for results regarding compatible systems of representatives (see [K4] for more details).

In this paper we want to bring together the results regarding constructions of (poly)matroids and generalized polymatroids (what are polyhedra bounded by sub- and supermodular functions satisfying an additional condition) showing that the majority of them can be expressed in the framework of the *partial intersection theorem* which says the following. Suppose $\mathbb{P}_1$ and $\mathbb{P}_2$ are generalized polymatroids in $\mathbb{R}^n \oplus \mathbb{R}^m$ and $\mathbb{R}^m$, respectively. Then there exists a generalized polymatroid $\mathbb{P}$ in $\mathbb{R}^n$ such that an n-dimensional vector $\mathbf{u}$ is from $\mathbb{P}$ iff there exists an m-dimensional vector $\mathbf{v}$ from $\mathbb{P}_2$ so that the direct sum of $\mathbf{u}$ and $\mathbf{v}$ is from $\mathbb{P}_1$.

In Section 4 an *abstract generalized polymatroidal network flow* model is introduced. It is equivalent with the flow models from [K6], Hassin [Ha], Lawler and Martel [LM1], [LM3], and generalizes the classical flow model of Ford and Fulkerson [FF]. Moreover, it has plenty of common features with the concept of group-valued

*Received by the editors December 13, 1995; accepted for publication (in revised form) July 31, 1997; published electronically July 7, 1998. This paper was finished during a JSPS fellowship at Keio University, Yokohama, Japan, and was partially supported by the Monbusho's Grant-in-Aid.
http://www.siam.org/journals/sidma/11-3/29598.html
†MÚ SAV, Štefánikova 49, 814 73 Bratislava, Slovakia (kochol@savba.sk).

flows presented in [K7] and [K8]. Thus, in a certain sense, our model unifies the concept of classical flows with the concept of group-valued flows.

The new flow model is used in Theorem 5, which is equivalent to the partial intersection theorem, and says the following. Let $G = (V(G), E(G))$ be a directed graph with a fixed vertex t so that with any arc x directed from u to v it contains an arc $-x$ directed from v to u and $-(-x) = x$. A flow in G is a mapping $f : E(G) \to \mathbb{R}$ so that for any arc x, $f(x) = -f(-x)$, and for any vertex $v \in V(G) \setminus \{t\}$, the restriction of f on the set of arcs directed into v is in a given generalized polymatroid. Then the restrictions of the flows on the set of arcs directed into t form a generalized polymatroid. Similar results, Theorems 6 and 7, are presented in section 4. They are formulated for flow models from Lawler and Martel [LM1], [LM3].

In the framework of Theorems 5, 6, and 7 the following operations on (generalized) polymatroids and matroids can very easily be described: sum, discrete sum, translation, dual, **c**-dual, restriction, truncation, (inverse) homomorphic image, and intersections with a plank and a box. This will be discussed later in section 6. In section 7, it is shown that Theorems 5, 6, and 7 generalize the theorem of Edmonds and Fulkerson [EF] (saying that partial transversals of a finite system of finite sets form a matroid), the constructions of gammoids from Perfect [Pe] and Pym [Py], the linking systems of Schrijver [S2], and other results from transversal theory.

Some new results are presented in the last section. Primarily, we characterize the polyhedron composed from the restrictions of all feasible flows on a fixed edge-cut in the flow model from section 4. A similar characterization can be obtained for the flow models from [Ha], [K6], [LM1], [LM3], and the classical flow model of Ford and Fulkerson [FF].

As was pointed out previously, many operations on generalized polymatroids can be described by a flow network. From the structure of this network we can determine whether the operation gives the resulting polyhedron equal to a generalized or a quasi polymatroid (which arises from a generalized polymatroid after reflexion of some of the coordinates). An example of the latter case is the following: Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be generalized polymatroids in $\mathbb{R}^n \oplus \mathbb{R}^m$ and $\mathbb{R}^m \oplus \mathbb{R}^k$, respectively. Take $\mathbb{Q}$ to be the set of the direct sums $\mathbf{u} \oplus \mathbf{v}$ where $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^k$ and there exists $\mathbf{w} \in \mathbb{R}^{\mathbf{m}}$ so that $\mathbf{u} \oplus \mathbf{w}$ is from $\mathbb{P}_1$ and $\mathbf{w} \oplus \mathbf{v}$ is from $\mathbb{P}_2$. Then $\mathbb{Q}$ is a quasi polymatroid in $\mathbb{R}^n \oplus \mathbb{R}^k$, but, in general, no generalized polymatroid.

2. Preliminaries. Throughout this paper, let $\mathbb{R}^S$ ($\mathbb{Z}^S$) denote the collection of the real (integer)-valued vectors indexed by a finite set S. For each $\mathbf{u} \in \mathbb{R}^S$ and $s \in S$ denote the sth coordinate of $\mathbf{u}$ by $\mathbf{u}(s)$. If $\mathbf{u} \in \mathbb{R}^S$ and $X \subseteq S$, $\mathbf{u}(X)$ is defined to be $\sum_{s \in X} \mathbf{u}(s)$, and $\mathbf{u}|X$ denotes the restriction of $\mathbf{u}$ to X. For two vectors $\mathbf{u} \in \mathbb{R}^S$ and $\mathbf{u}' \in \mathbb{R}^{S'}$ with $S \cap S' = \emptyset$, their *direct sum* $\mathbf{u} \oplus \mathbf{u}' \in \mathbb{R}^{S \cup S'}$ is defined by

$$(\mathbf{u} \oplus \mathbf{u}')(s) = \begin{cases} \mathbf{u}(s) & \text{if } s \in S, \\ \mathbf{u}'(s) & \text{if } s \in S'. \end{cases}$$

Clearly, $(\mathbf{u} \oplus \mathbf{u}')|S = \mathbf{u}$ and $(\mathbf{u} \oplus \mathbf{u}')|S' = \mathbf{u}'$. For convenience, we suppose that $\mathbb{R}^\emptyset = \{\emptyset\}$, $\mathbf{u} \oplus \emptyset = \mathbf{u}$, $\mathbf{u}|\emptyset = \emptyset$, and $\mathbf{u}(\emptyset) = 0$.

If it is clear from the context that we are referring to a set rather than to an element we abbreviate $\{x\}$ to x. For example, $X \cup x$ means $X \cup \{x\}$ and $\rho(x)$ means $\rho(\{x\})$.

Let $\rho\, 2^S \to \mathbb{R} \cup \{\infty\}$, $\sigma\, 2^S \to \mathbb{R} \cup \{-\infty\}$ so that $\rho(\emptyset) = \sigma(\emptyset) = 0$ and for any

$X, Y \subseteq S$,

$$\rho(X) + \rho(Y) \geq \rho(X \cup Y) + \rho(X \cap Y), \tag{1}$$

$$\sigma(X) + \sigma(Y) \leq \sigma(X \cup Y) + \sigma(X \cap Y), \tag{2}$$

$$\rho(X) - \sigma(Y) \geq \rho(X \setminus Y) - \sigma(Y \setminus X). \tag{3}$$

(Equations (1) and (2) state that ρ and σ are *submodular* and *supermodular*, respectively, and (3) states that ρ and σ are *compliant*.) Then the set (see [Fr], [Kov], [KP])

$$\mathbb{P} = \{\mathbf{u} \in \mathbb{R}^S; \sigma(X) \leq \mathbf{u}(X) \leq \rho(X) \text{ for every } X \subseteq S\}$$

is called a *g-polymatroid (generalized polymatroid)* on the *ground set* S. Formally, we write $\mathbb{P} = (S, \rho, \sigma)$. If both ρ and σ are integer valued (i.e., are mappings to $\mathbb{Z} \cup \{\infty\}$ and $\mathbb{Z} \cup \{-\infty\}$, respectively), then $\mathbb{P}$ is called *integral*.

If $\sigma \equiv 0$, then by (3), ρ is monotone and nonnegative (i.e., $0 \leq \rho(X) \leq \rho(Y)$ if $X \subseteq Y \subseteq S$) and $\mathbb{P}$ is called a *polymatroid*. Moreover, if $\rho(s) = 0$ or 1 for any $s \in S$ and ρ is integral, then $\mathbb{P}$ is a *matroid*. If $\sigma(X) = -\infty$ for any $\emptyset \neq X \subseteq S$, we get an *extended polymatroid* (see [GLS] or [S4]). If $\mathbb{P}$ is matroid or polymatroid, then we formally write $\mathbb{P} = (S, \rho)$. Note that matroid (S, ρ) is usually identified with the system of sets $\{X \subseteq S; \rho(X) = |X|\}$ (see, e.g., [We], [A], [R], [NW]).

For convenience, we consider $\{\emptyset\} = 2^\emptyset$ to be the g-polymatroid on $\emptyset$, i.e., $\{\emptyset\} = (\emptyset, \rho_\emptyset, \sigma_\emptyset)$ where, by definition, $\rho_\emptyset(\emptyset) = \sigma_\emptyset(\emptyset) = 0$.

Generalized polymatroids present a natural extension of polymatroids and preserve a majority of their nice properties. They have been introduced independently by Frank [Fr] and Kovalev [Kov] (see also [KP]). A detailed study of them can be found in the survey article of Frank and Tardos [FT]. We now recall some basic results from it. For instance, any nontrivial (integral) g-polymatroid contains an (integral) vector. Moreover, for any $X \subseteq S$,

$$\rho(X) = \max\{\mathbf{u}(X);\ \mathbf{u} \in \mathbb{P}\}, \tag{4}$$

$$\sigma(X) = \min\{\mathbf{u}(X);\ \mathbf{u} \in \mathbb{P}\}. \tag{5}$$

Note that in this paper we follow the usual min-max notation and suppose that if a subset of $\mathbb{R}$ is not bounded above (below), then its maximum (minimum) is ∞ ($-\infty$).

The next theorem is equivalent to the Edmonds intersection theorem (see [Fr], [S3]).

THEOREM 1. *Let $\mathbb{P}_1 = (S, \rho_1, \sigma_1)$ and $\mathbb{P}_2 = (S, \rho_2, \sigma_2)$ be two g-polymatroids. Then the following linear system is totally dual integral:*

$$\sigma_i(X) \leq \mathbf{u}(X) \leq \rho_i(X) \quad (i = 1, 2,\ X \subseteq S). \tag{6}$$

Let us recall that a system $\mathbf{Ax} \leq \mathbf{b}$ of inequalities is *totally dual integral* (*TDI*) if the minimum in the linear programming duality equation

$$\max\{\mathbf{wx};\ \mathbf{Ax} \leq \mathbf{b}\} = \min\{\mathbf{yb};\ \mathbf{y} \geq 0, \mathbf{yA} = \mathbf{w}\} \tag{7}$$

has an integral optimum solution for each integral objective function $\mathbf{w}$ for which the minimum exists. Hoffman [Ho] and Edmonds and Giles [EG] showed that if $\mathbf{Ax} \leq \mathbf{b}$ is TDI and $\mathbf{b}$ is integral, then the maximum in (7) also has an integral optimum solution. Therefore, if $\mathbb{P}_1$ and $\mathbb{P}_2$ are integral g-polymatroids on S, then $\mathbb{P}_1 \cap \mathbb{P}_2$ is

an integral polyhedron, i.e., any of its faces contains an integral vector. Note that the TDI property regards the linear system and not the polyhedron defined by this system. See, e.g., [S5] for more details.

System (6) remains TDI if ρ_i and σ_i are defined on intersecting families (see [L1], [L2], [S3], [S4], [Fr], [FT]). But we do not follow this approach because we deal with constructions of polyhedra and not with TDI systems.

From Theorem 1 are the following corollaries (see, e.g., [K6], [FT]).

COROLLARY 1. *Let $\mathbb{P}_1 = (S, \rho_1, \sigma_1)$ and $\mathbb{P}_2 = (S, \rho_2, \sigma_2)$ be two (integral) g-polymatroids. Then they have an (integral) vector in common iff $\rho_1(X) \geq \sigma_2(X)$ and $\rho_2(X) \geq \sigma_1(X)$ for any $X \subseteq S$.*

COROLLARY 2. *Let $\mathbb{P}_1 = (S, \rho_1, \sigma_1)$ and $\mathbb{P}_2 = (S, \rho_2, \sigma_2)$ be two g-polymatroids having a vector in common. Then,*

$$\max\{\mathbf{u}(S); \mathbf{u} \in \mathbb{P}_1 \cap \mathbb{P}_2\} = \min_{X \subseteq S} (\rho_1(X) + \rho_2(S \setminus X)),$$
$$\min\{\mathbf{u}(S); \mathbf{u} \in \mathbb{P}_1 \cap \mathbb{P}_2\} = \max_{X \subseteq S} (\sigma_1(X) + \sigma_2(S \setminus X)).$$

Furthermore, if $\mathbb{P}_1$ and $\mathbb{P}_2$ are integral, then the maximal and minimal values of $\mathbf{u}(S)$ can be obtained (if they are finite) for integral vectors.

If $\mathbb{P}_1 \subseteq \mathbb{R}^{S_1}$, then denote $-\mathbb{P}_1 = \{\mathbf{u} \in \mathbb{R}^{S_1}; -\mathbf{u} \in \mathbb{P}_1\}$. Further, if $\mathbb{P}_2 \subseteq \mathbb{R}^{S_2}$ and $S_1 \cap S_2 = \emptyset$, then the *direct sum* of $\mathbb{P}_1$ and $\mathbb{P}_2$ is defined as $\mathbb{P}_1 \oplus \mathbb{P}_2 = \{\mathbf{u} \oplus \mathbf{v}; \mathbf{u} \in \mathbb{P}_1, \mathbf{v} \in \mathbb{P}_2\}$. Clearly, $-(-\mathbb{P}_1) = \mathbb{P}_1$ and $\mathbb{P}_1 \oplus \{\emptyset\} = \mathbb{P}_1$. Furthermore, if $\mathbb{P}_1 = (S_1, \rho_1, \sigma_1)$ and $\mathbb{P}_2 = (S_2, \rho_2, \sigma_2)$ are (integral) g-polymatroids, then so are $-\mathbb{P}_1 = (S_1, -\sigma_1, -\rho_1)$ and $\mathbb{P}_1 \oplus \mathbb{P}_2 = (S_1 \cup S_2, \rho, \sigma)$, where $\rho(X_1 \cup X_2) = \rho_1(X_1) + \rho_2(X_2)$ and $\sigma(X_1 \cup X_2) = \sigma_1(X_1) + \sigma_2(X_2)$ for any $X_1 \subseteq S_1$, $X_2 \subseteq S_2$.

Example 1. Let $\rho_\infty(\emptyset) = \sigma_\infty(\emptyset) = 0$ and $\rho_\infty(X) = \infty$, $\sigma_\infty(X) = -\infty$ for any $\emptyset \neq X \subseteq S$. Then $(S, \rho_\infty, \sigma_\infty) = \mathbb{R}^S$ is called the *free g-polymatroid* on S. Similarly, the polymatroid (S, ρ_∞) is called the *free polymatroid* on S. It contains the vectors from $\mathbb{R}^S$ with nonnegative coordinates.

Example 2. Let $\overline{\mathbb{P}} = (\{a, b\}, \overline{\rho}, \overline{\sigma})$ be a g-polymatroid such that $\overline{\rho}(\{a, b\}) = \overline{\sigma}(\{a, b\}) = 0$ and $\overline{\rho}(x) = \overline{\sigma}(x) = \infty$ for $x = a, b$. $\overline{\mathbb{P}}$ is called the *principal g-polymatroid* on $\{a, b\}$. Clearly, $\overline{\mathbb{P}} = \{\mathbf{u} \in \mathbb{R}^{\{a,b\}}; \mathbf{u}(a) = -\mathbf{u}(b)\}$.

Example 3. If $\mathbf{u} \in \mathbb{R}^S$, then $\mathbb{P}_\mathbf{u} = \{\mathbf{u}\}$ is a g-polymatroid $(S, \rho_\mathbf{u}, \sigma_\mathbf{u})$ so that $\rho_\mathbf{u}(X) = \sigma_\mathbf{u}(X) = \mathbf{u}(X)$ for any $X \subseteq S$.

Two g-polymatroids $\mathbb{P} = (S, \rho, \sigma)$ and $\mathbb{P}' = (S', \rho', \sigma')$ are called *isomorphic* if there exists a bijection $\varphi : S \to S'$ such that for any $X \subseteq S$, $\rho(X) = \rho'(\varphi(X))$ and $\sigma(X) = \sigma'(\varphi(X))$.

3. Partial intersection theorem.

THEOREM 2. *Let S, T be finite disjoint sets and $\mathbb{P}_1 = (S \cup T, \rho_1, \sigma_1)$, $\mathbb{P}_2 = (T, \rho_2, \sigma_2)$ be (integral) g-polymatroids. Suppose $\rho_1(Y) \geq \sigma_2(Y)$, $\rho_2(Y) \geq \sigma_1(Y)$ for any $Y \subseteq T$. Then there exists an (integral) g-polymatroid $\mathbb{P} = (S, \rho, \sigma)$ such that for any $X \subseteq S$,*

$$\rho(X) = \min_{Y \subseteq T} \big(\rho_1(X \cup Y) - \sigma_2(Y)\big), \tag{8}$$

$$\sigma(X) = \max_{Y \subseteq T} \big(\sigma_1(X \cup Y) - \rho_2(Y)\big), \tag{9}$$

and an (integral) vector $\mathbf{u} \in \mathbb{R}^S$ is from $\mathbb{P}$ iff there exists an (integral) vector $\mathbf{v} \in \mathbb{P}_2$ so that $\mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1$. $\mathbb{P}$ is called the partial intersection of $\mathbb{P}_1$ and $\mathbb{P}_2$.

Proof. Let ρ, σ be the functions defined by (8), (9), respectively, and $X, X' \subseteq S$. Choose $Y, Y', Y'' \subseteq T$ so that $\rho(X) = \rho_1(X \cup Y) - \sigma_2(Y)$, $\rho(X') = \rho_1(X' \cup Y') - \sigma_2(Y')$, and $\sigma(X') = \sigma_1(X' \cup Y'') - \rho_2(Y'')$. Then using compliance, sub- and supermodularity of ρ_i, σ_i, and (8), (9) we get

$$\begin{aligned}
\rho(X) + \rho(X') &= \rho_1(X \cup Y) - \sigma_2(Y) + \rho_1(X' \cup Y') - \sigma_2(Y') \\
&\geq \rho_1(X \cup X' \cup Y \cup Y') + \rho_1((X \cap X') \cup (Y \cap Y')) - \sigma_2(Y \cup Y') - \sigma_2(Y \cap Y') \\
&\geq \rho(X \cup X') + \rho(X \cap X'), \\
\rho(X) - \sigma(X') &= \rho_1(X \cup Y) - \sigma_2(Y) - \sigma_1(X' \cup Y'') + \rho_2(Y'') \\
&\geq \rho_1((X \setminus X') \cup (Y \setminus Y'')) - \sigma_1((X' \setminus X) \cup (Y'' \setminus Y)) - \sigma_2(Y \setminus Y'') + \rho_2(Y'' \setminus Y) \\
&\geq \rho(X \setminus X') - \sigma(X' \setminus X).
\end{aligned}$$

Thus (1) and (3) are satisfied for any $X, X' \subseteq S$ and (2) can be verified similarly as (1). Moreover, $\rho(\emptyset) = \sigma(\emptyset) = 0$ because $\rho_1(Y) \geq \sigma_2(Y)$, $\rho_2(Y) \geq \sigma_1(Y)$ for any $Y \subseteq T$. Therefore, $\mathbb{P} = (S, \rho, \sigma)$ is a g-polymatroid. By (8) and (9), if $\mathbb{P}_1$ and $\mathbb{P}_2$ are integral, so then is $\mathbb{P}$.

If $\mathbf{v} \in \mathbb{P}_2$ and $\mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1$, then for any $X \subseteq S$, $Y \subseteq T$, $\mathbf{u}(X) + \mathbf{v}(Y) \leq \rho_1(X \cup Y)$, $\mathbf{v}(Y) \geq \sigma_2(Y)$, i.e., $\mathbf{u}(X) \leq \rho_1(X \cup Y) - \sigma_2(Y)$. Thus, $\mathbf{u}(X) \leq \rho(X)$ $(X \subseteq S)$. Similarly, $\mathbf{u}(X) \geq \sigma(X)$ $(X \subseteq S)$ and, therefore, $\mathbf{u} \in \mathbb{P}$.

For the converse, we prove that if $\mathbf{u} \in \mathbb{P}$, then there exists $\mathbf{v} \in \mathbb{P}_2$ so that $\mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1$ and, moreover, if $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbf{u}$ are integral, then $\mathbf{v}$ can be chosen to be integral too. We shall do it in two steps.

(a) Suppose $\mathbb{P}_2 = \{\mathbf{v}\}$ (i.e., $\mathbf{v} \in \mathbb{R}^T$ and $\rho_2(Y) = \sigma_2(Y) = \mathbf{v}(Y)$ for any $Y \subseteq T$). If $\mathbf{u} \in \mathbb{P}$, then, by (8), for any $X \subseteq S$, $Y \subseteq T$, $\mathbf{u}(X) \leq \rho(X) \leq \rho_1(X \cup Y) - \mathbf{v}(Y)$ and, thus, $\mathbf{u}(X) + \mathbf{v}(Y) \leq \rho_1(X \cup Y)$. Similarly, $\mathbf{u}(X) + \mathbf{v}(Y) \geq \sigma_1(X \cup Y)$ and, therefore, $\mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_{\mathbf{1}}$, which proves the theorem in this case.

(b) Suppose $\mathbb{P}_2$ is arbitrary and $\mathbf{u} \in \mathbb{P}$. Let $\overline{\mathbb{P}}_{\mathbf{u}} = \{\mathbf{v} \in \mathbb{R}^T; \mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1\}$. From item (a) it follows that $\overline{\mathbb{P}}_{\mathbf{u}}$ is a g-polymatroid $(T, \overline{\rho}_{\mathbf{u}}, \overline{\sigma}_{\mathbf{u}})$ so that $\overline{\rho}_{\mathbf{u}}(Y) = \min_{X \subseteq S}(\rho_1(X \cup Y) - \mathbf{u}(X))$ and $\overline{\sigma}_{\mathbf{u}}(Y) = \max_{X \subseteq S}(\sigma_1(X \cup Y) - \mathbf{u}(X))$ for any $Y \subseteq T$ (note that we must "interchange" the role of S and T).

Fix $Y \subseteq T$ and choose $X \subseteq S$ so that $\overline{\rho}_{\mathbf{u}}(Y) = \rho_1(X \cup Y) - \mathbf{u}(X)$. Since $\mathbf{u} \in \mathbb{P}$, we have $\mathbf{u}(X) \leq \rho(X)$ and, by (8), $\rho(X) \leq \rho_1(X \cup Y) - \sigma_2(Y)$. Thus

$$\begin{aligned}
\overline{\rho}_{\mathbf{u}}(Y) &= \rho_1(X \cup Y) - \mathbf{u}(X) \geq \rho_1(X \cup Y) - \rho(X) \\
&\geq \rho_1(X \cup Y) - \rho_1(X \cup Y) + \sigma_2(Y) = \sigma_2(Y).
\end{aligned}$$

Similarly, $\overline{\sigma}_{\mathbf{u}}(Y) \leq \rho_2(Y)$. Therefore, for any $Y \subseteq T$, $\overline{\rho}_{\mathbf{u}}(Y) \geq \sigma_2(Y)$ and $\overline{\sigma}_{\mathbf{u}}(Y) \leq \rho_2(Y)$ and, by Corollary 1, $\overline{\mathbb{P}}_{\mathbf{u}}$ and $\mathbb{P}_2$ have a vector $\mathbf{v}$ in common. Thus, by the definition of $\overline{\mathbb{P}}_{\mathbf{u}}$, we have $\mathbf{v} \in \mathbb{P}_2$ so that $\mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1$.

Furthermore, if $\mathbb{P}_1$ and $\mathbf{u}$ are integral, so then is $\overline{\mathbb{P}}_{\mathbf{u}}$. If $\mathbb{P}_2$ is also integral, then, by Corollary 1, $\mathbf{v}$ can be chosen to be integral, concluding the proof. □

In the proof, we have used only Corollary 1 and the inequalities (1)–(3), though we are aware that some known results can be used effectively too. (For instance, $\overline{\mathbb{P}}_{\mathbf{u}}$ can be obtained after intersecting $\mathbb{P}_1$ with a suitable box and then applying restriction to T (see Example 4 and [FT]). Similarly, $\mathbb{P}$ can be obtained.) But the aim of this paper is not to prove Theorem 2, but to introduce it as a general framework for constructions and operations. Therefore, we did not use the results which should be later presented as consequences of this theorem.

The next example is simple and transparent.

Example 4. Let $\mathbb{P} = (S, \rho, \sigma)$ be a g-polymatroid and $X \subseteq S$. Then the partial intersection of $\mathbb{P}$ and $\mathbb{R}^{S \setminus X}$ is called the *restriction* (or *projection*) of $\mathbb{P}$ to X and denoted by $\mathbb{P}|X = (X, \rho|X, \sigma|X)$. By Theorem 2 and Example 1, $\rho|X$ ($\sigma|X$) is a restriction of ρ (σ) to 2^X and $\mathbb{P}|X = \{\mathbf{u}|X; \mathbf{u} \in \mathbb{P}\}$.

4. Flows in abstract networks. All graphs displayed in this paper are finite and may have multiple edges but no loops. Each edge gives rise to two oppositely directed edges called *arcs*. For an arc x, we denote by $-x$ the *reverse* arc arising from the same edge. The set of all arcs of a graph $G = (V(G), E(G))$ will be denoted by $D(G)$, i.e., $|D(G)| = 2|E(G)|$. For any $U \subseteq V(G)$, Δ_U denotes the set of arcs directed from $V(G) \setminus U$ to U (we write Δ_v if $U = \{v\}$).

An *abstract g-polymatroidal flow network* $\mathcal{N}$ (*abstract network*) is a graph G where each vertex v of G is accompanied with a polymatroid $\mathbb{P}_v = (\Delta_v, \rho_v, \sigma_v)$. $\mathcal{N}$ is called *integral* if any $\mathbb{P}_v$ is integral. A *chain* in $\mathcal{N}$ is any $f \in \mathbb{R}^{D(G)}$ satisfying $f(-x) = -f(x)$ for any arc x of G. A chain f in $\mathcal{N}$ is called *flow* in $\mathcal{N}$ if for any vertex v of G, $f|\Delta_v \in \mathbb{P}_v$, i.e., $\sigma_v(X) \le f(X) \le \rho_v(X)$ for any $X \subseteq \Delta_v$. If a chain (flow) in $\mathcal{N}$ is integer valued, then it is called *integral*. A *U-value* of a chain f is $f(\Delta_U)$ for any $U \subseteq V(G)$. A vertex v of G is called *inner* if $\rho_v(\Delta_v) = \sigma_v(\Delta_v) = 0$. Otherwise, it is called *outer*.

Suppose $X \subseteq D(G)$. Then, $-X = \{x; -x \in X\}$. X is called *symmetric* (*asymmetric*) if $-X = X$ ($-X \cap X = \emptyset$). If $A \subseteq E(G)$, then $D(A)$ denotes the set of arcs rising from the edges of A (for instance, $D(E(G)) = D(G)$ and $D(e)$ is the couple of arcs arising from an edge e of G). If $U \subseteq V(G)$, then $-U = V(G) \setminus U$ (i.e., $-\Delta_U = \Delta_{-U}$). By a *U-cut* of $\mathcal{N}$ we mean a triple (U, A, B) so that $A = A' \setminus \Delta_{-U}$, $B = B' \setminus \Delta_U$, where the couple A', B' is a partition of $D(G)$ into two symmetric sets. The *upper capacity* of the U-cut (U, A, B) is defined as

$$c_{\mathrm{up}}(U, A, B) = \sum_{v \in U} \rho_v(\Delta_v \cap A) - \sum_{v \in -U} \sigma_v(\Delta_v \cap B).$$

The *lower capacity* of the U-cut (U, A, B) is defined

$$c_{\mathrm{low}}(U, A, B) = \sum_{v \in U} \sigma_v(\Delta_v \cap A) - \sum_{v \in -U} \rho_v(\Delta_v \cap B).$$

Clearly, $c_{\mathrm{up}}(U, A, B) = -c_{\mathrm{low}}(-U, B, A)$. Note that we allow U, A, or B to be empty.

The next theorem characterizes the abstract networks admitting flows. Theorem 4 is the max-flow min-cut theorem for our model.

THEOREM 3. *Let $\mathcal{N}$ be an (integral) abstract network on a graph G. Then the following conditions are equivalent:*

(a) *$\mathcal{N}$ admits an (integral) flow.*

(b) *Every $V(G)$- and $\emptyset$-cut of $\mathcal{N}$ has nonnegative upper capacity.*

(c) *Every $V(G)$- and $\emptyset$-cut of $\mathcal{N}$ has nonpositive lower capacity.*

Proof. For any $e \in E(G)$, let $\mathbb{P}_e = (D(e), \rho_e, \sigma_e)$ be the principal g-polymatroid on $D(e)$. Take $\mathbb{P}_1 = (D(G), \rho_1, \sigma_1) = \bigoplus_{e \in E(G)} \mathbb{P}_e$ and $\mathbb{P}_2 = (D(G), \rho_2, \sigma_2) = \bigoplus_{v \in V(G)} \mathbb{P}_v$. Then f is a flow in $\mathcal{N}$ iff $f \in \mathbb{P}_1 \cap \mathbb{P}_2$. By Corollary 1, the intersection of $\mathbb{P}_1$ and $\mathbb{P}_2$ is nonempty iff $\rho_1(X) \ge \sigma_2(X)$ and $\rho_2(X) \ge \sigma_1(X)$ for any $X \subseteq D(G)$. Trivially this holds if $-X \ne X$, because then $\infty = \rho_1(X) = -\sigma_1(X)$ (see Example 2). Thus, these conditions remain to verify only for symmetric X. But then $\rho_1(X) = \sigma_1(X) = 0$, and, therefore, (a), (b), and (c) are equivalent. □

THEOREM 4. *Let $\mathcal{N}$ be an abstract network on a graph G admitting a flow and $U \subseteq V(G)$. Then the maximal (minimal) U-value of a flow in $\mathcal{N}$ is equal to the*

minimal upper (maximal lower) capacity of a U-cut in $\mathcal{N}$. Furthermore, if $\mathcal{N}$ is integral and the maximal (minimal) U-value of $\mathcal{N}$ is finite, then there exists integral flow in $\mathcal{F}$ with the maximal (minimal) U-value.

Proof. For any $v \in -U$, let $\overline{\mathbb{P}}_v$ be the g-polymatroid on $-\Delta_v$ isomorphic with $-\mathbb{P}_v$ under the isomorphism $x \mapsto -x$, and for any $e \in E(G)$, let $\mathbb{P}_e$ be the principal g-polymatroid on $D(e)$. Let E_U and E_{-U} be the sets comprising the edges with both ends in U and $-U$, respectively. Take $D' = D(G) \setminus \Delta_{-U}$ and

$$\overline{\mathbb{P}}_1 = (D', \overline{\rho}_1, \overline{\sigma}_1) = \left(\bigoplus_{e \in E_{-U}} \mathbb{P}_e\right) \oplus \left(\bigoplus_{v \in U} \mathbb{P}_v\right),$$

$$\overline{\mathbb{P}}_2 = (D', \overline{\rho}_2, \overline{\sigma}_2) = \left(\bigoplus_{e \in E_U} \mathbb{P}_e\right) \oplus \left(\bigoplus_{v \in -U} \overline{\mathbb{P}}_v\right).$$

Then f is a flow in $\mathcal{N}$ iff $\overline{f} = f|D' \in \overline{\mathbb{P}}_1 \cap \overline{\mathbb{P}}_2$ and the U-value of f is equal to $\overline{f}(D')$. Therefore, by Corollary 2, the maximal U-value of a flow in $\mathcal{N}$ is equal to $\min_{X \subseteq D'}(\overline{\rho}_1(X) + \overline{\rho}_2(D' \setminus X))$. Using the arguments from the proof of Theorem 3, we can show that $X \setminus \Delta_U$ must be symmetric and that $\overline{\rho}_1(X) + \overline{\rho}_2(D' \setminus X) = c_{\mathrm{up}}(U, X, D' \setminus X)$. Thus the maximal U-value is equal to the minimal upper capacity of a U-cut. Similarly, the property for the minimal U-value can be checked. The conditions for integrality follows from Corollary 2. □

Suppose $\mathcal{N}$ is an abstract network on a graph G, $U \subseteq V(G)$ and f is a flow in $\mathcal{N}$. Then $f|\Delta_U$ is called a *U-transversal* of $\mathcal{N}$. The set of all U-transversals of $\mathcal{N}$ is called a *U-gammoid* of $\mathcal{N}$. (If $U = \{v\}$, then we speak about a *v-transversal* and a *v-gammoid* of $\mathcal{N}$.) In section 7 we show that these notions generalize transversals and gammoids, which are known from transversal theory.

THEOREM 5. *Let $\mathcal{N}$ be an (integral) abstract network on a graph G with a collection of g-polymatroids $\mathbb{P}_v = (\Delta_v, \rho_v, \sigma_v)$ ($v \in V(G)$). Suppose $\mathcal{F}$ admits a flow and has an outer vertex t such that $\mathbb{P}_t = \mathbb{R}^{\Delta_t}$. Let $G \setminus t$ denote the graph obtained from G after removing the vertex t and all its neighboring edges. Then the t-gammoid of $\mathcal{N}$ is an (integral) g-polymatroid $\mathbb{P} = (\Delta_t, \rho, \sigma)$ such that*

$$\begin{aligned}\rho(X) &= \max\{f(X);\ f \text{ is an (integral) flow in } \mathcal{N}\}\\ &= \min_{Z \subseteq E(G \setminus t)} \sum_{v \in V(G \setminus t)} -\sigma_v(\Delta_v \cap (-X \cup D(Z))),\end{aligned}$$

$$\begin{aligned}\sigma(X) &= \min\{f(X);\ f \text{ is an (integral) flow in } \mathcal{N}\}\\ &= \max_{Z \subseteq E(G \setminus t)} \sum_{v \in V(G \setminus t)} -\rho_v(\Delta_v \cap (-X \cup D(Z)))\end{aligned}$$

for any $X \subseteq \Delta_t$. Furthermore, if $\mathcal{N}$ is integral, then any integral t-transversal of $\mathcal{N}$ can be extended into an integral flow in $\mathcal{N}$.

Proof. Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be defined as in the proof of Theorem 3. Take $\mathbb{P}'_2 = (D(G) \setminus \Delta_t, \rho'_2, \sigma'_2) = \bigoplus_{v \in V(G) \setminus t} \mathbb{P}_v = \mathbb{P}_2|(D(G) \setminus \Delta_t)$. Then the t-gammoid of $\mathcal{N}$ is the partial intersection of $\mathbb{P}_1$ and $\mathbb{P}'_2$, and by Theorem 2, it is equal to $\mathbb{P} = (\Delta_t, \rho, \sigma)$ so that for any $X \subseteq \Delta_t$,

$$\rho(X) = \min_{Y \subseteq D(G) \setminus \Delta_t} (\rho_1(X \cup Y) - \sigma'_2(Y)).$$

Also, from the properties of $\mathbb{P}_1$ and Example 2, it follows that we should consider only the cases if $X \cup Y$ is symmetric, i.e., $Y = -X \cup D(Z)$ where $Z \subseteq E(G \setminus t)$. Then the

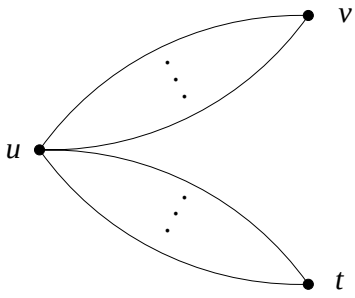


FIG. 1.

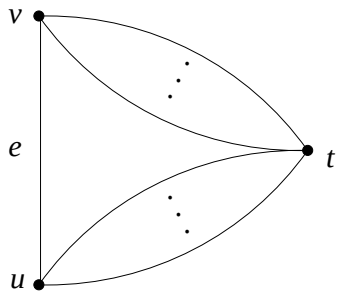


FIG. 2.

formula for $\rho(X)$ from Theorem 5 is valid. A similar situation holds for $\sigma(X)$. The conditions for integrality follow from Theorem 2. □

Take the abstract network $\mathcal{N}_1$ from Fig. 1 so that $\mathbb{P}_u$, $\mathbb{P}_v$, $\mathbb{P}_t$ are isomorphic with $-\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{R}^{\Delta_t}$, respectively. Then the t-gammoid of $\mathcal{N}_1$ is just the partial intersection of $\mathbb{P}_1$ and $\mathbb{P}_2$, and Theorem 2 follows from Theorem 5. Thus these two statements are equivalent.

Examples 5 and 6 will be used in the next section. The latter presents an application of Theorem 5.

Example 5. Let $\mathbb{P} = (S, \rho, \sigma)$ be a g-polymatroid and $S' = S \cup s'$, $s' \notin S$. Define $\rho' : 2^{S'} \to \mathbb{R} \cup \{\infty\}$, $\sigma' : 2^{S'} \to \mathbb{R} \cup \{-\infty\}$ such that

$$\begin{aligned} \rho'(X) = \rho(X), \quad \sigma'(X) = \sigma(X) \quad \text{if} \ \ X \subseteq S, \\ \rho'(X) = -\sigma(S \setminus X), \quad \sigma'(X) = -\rho(S \setminus X) \quad \text{if} \ \ s' \in X \subseteq S'. \end{aligned} \tag{10}$$

By (1)–(3), $\mathbb{P}' = (S', \rho', \sigma')$ is a g-polymatroid. We call it the 0-*extension* of $\mathbb{P}$ to S'. Clearly, $\mathbb{P}' = \{\mathbf{u} \in \mathbb{R}^{S'}; \mathbf{u}|S \in \mathbb{P} \text{ and } \mathbf{u}(s') = -\mathbf{u}(S)\}$ (see also Fujishige [F2, Theorem 3.58]).

Example 6. Let $\mathbb{P}_1 = (S_1, \rho_1, \sigma_1)$, $\mathbb{P}_2 = (S_2, \rho_2, \sigma_2)$ be g-polymatroids, $S_1 \cap S_2 = \emptyset$, $\rho_1(S_1) \geq \sigma_2(S_2)$ and $\rho_2(S_2) \geq \sigma_1(S_1)$. Then there exists a g-polymatroid $\mathbb{P}_1 \mathbin{\dot{\cup}} \mathbb{P}_2 = (S_1 \cup S_2, \rho, \sigma)$ so that $\mathbb{P}_1 \mathbin{\dot{\cup}} \mathbb{P}_2 = \{\mathbf{u} \oplus \mathbf{v}; \mathbf{u} \in \mathbb{P}_1, \mathbf{v} \in -\mathbb{P}_2, \mathbf{u}(S_1) = -\mathbf{v}(S_2)\}$ and for any $X \subseteq S_1$, $Y \subseteq S_2$,

$$\begin{aligned} \rho(X \cup Y) = \min\{\rho_1(X) - \sigma_2(Y), -\sigma_1(S_1 \setminus X) + \rho_2(S_2 \setminus Y)\}, \\ \sigma(X \cup Y) = \max\{\sigma_1(X) - \rho_2(Y), -\rho_1(S_1 \setminus X) + \sigma_2(S_2 \setminus Y)\}. \end{aligned} \tag{11}$$

It can be obtained as follows. Take the edge e from the graph from Fig. 2 and denote by e_u and e_v the arcs arising from e and directed to u and v, respectively. Suppose

$S_1 = \Delta_u \setminus e_u$, $S_2 = \Delta_v \setminus e_v$. Consider the abstract network $\mathcal{N}_2$ on the graph from Fig. 2 so that $\mathbb{P}_u$ and $\mathbb{P}_v$ are the 0-extensions of $-\mathbb{P}_1$ and $\mathbb{P}_2$, respectively, and $\mathbb{P}_t = \mathbb{R}^{\Delta_t}$. Then the t-gammoid of $\mathcal{N}_2$ is isomorphic with $\mathbb{P}_1 \underset{=}{\ominus} \mathbb{P}_2$ and (11) follows from Theorem 5 and the equations in (10). Clearly, $\mathbb{P}_1 \underset{=}{\ominus} \mathbb{P}_2 = -(\mathbb{P}_2 \underset{=}{\ominus} \mathbb{P}_1)$.

5. Connections with other flow models. An *orientation* of a graph G is a directed graph (digraph) G' arising from G after endowing all of its edges by orientations. Then G is called the *underlying* graph of G'. Note that $E(G')$ is a maximal asymmetric subset of $D(G)$ (i.e., $E(G') \cap (-E(G')) = \emptyset$ and $E(G') \cup (-E(G')) = D(G)$) and $V(G) = V(G')$. If $U \subseteq V(G')$, then Δ_U^- denotes the set of arcs from $E(G')$ directed from U to $V(G') \setminus U$ and $\Delta_U^+ = \Delta_{V(G')\setminus U}^-$.

Suppose $\mathcal{N}$ is an abstract network in a graph G. If G' is an orientation of G, then any chain (flow) in $\mathcal{N}$ is uniquely determined by its values on the arcs of G'. Moreover, we can check that f is a chain (flow) in $\mathcal{N}$ iff $f' = f|E(G')$ is a flow (feasible flow) in a *quasi polymatroidal flow network* on G' (defined in [K6]). Thus these two flow models are equivalent. This fact gives meaning to several notions we have introduced before. For instance, if v is an inner vertex of the abstract network $\mathcal{N}$ (i.e., $\rho_v(\Delta_v) = \sigma_v(\Delta_v) = 0$) and f is a flow in $\mathcal{N}$, then the sum of the values of f' on the arcs from G' entering v is equal to the sum of the values of f' on the arcs from G' leaving v (in terms from [K6], v is *balanced*). This corresponds with the situation that the flow f' "enters and leaves" the network in the outer vertices and "comes through" the inner vertices. Furthermore, the U-value of f is equal to $f'(\Delta_U^+) - f'(\Delta_U^-)$. In this context it is clear that Theorem 4 is, in fact, the max-flow min-cut theorem for abstract networks.

Let us stress a similarity with the concept of group-valued flows on graphs as presented in [K7] and [K8]. In these papers, what we mean by an *abstract network* is a couple (G, S) where $S \subseteq V(G)$ is a set of *outer* vertices. The vertices from $V(G) \setminus S$ are called *inner*. Let A be an additive Abelian group. Then an *A-chain* in (G, S) is any mapping $\varphi\, D(G) \to A$ so that $\varphi(-x) = -\varphi(x)$ for any arc x of G. A *boundary* of φ is $\partial\varphi\, V(G) \to A$ so that $\partial\varphi(v) = \sum_{x \in -\Delta_v} \varphi(x)$. An A-chain is said to be an *A-flow* if $\partial\varphi(v) = 0$ for every inner vertex v of (G, S). Using the language of homology, the A-chains and A-flows in (G, S) correspond to 0-chains in G and to relative 1-cycles mod S with coefficients in A, respectively. We have used an analogical terminology for abstract g-polymatroidal flow networks and dealt with chains and flows (though following the notation used in combinatorial optimization we should call them "flows" and "feasible flows," respectively, as we have done in [K6]). Note that the A-flows in $(G, \emptyset)$ correspond with the usual definition of A-flows in G as presented, e.g., in the survey article of Jaeger [J] (in fact, it suffices to consider the restrictions of A-flows on an orientation of G).

Now we shall introduce the flow model from Lawler and Martel [LM3]. A *g-polymatroidal flow network* $\mathcal{F}$ is a digraph G' with a *source* s, a *sink* t, and a collection of g-polymatroids $\mathbb{P}_v^+ = (\Delta_v^+, \rho_v^+, \sigma_v^+)$, $\mathbb{P}_v^- = (\Delta_v^-, \rho_v^-, \sigma_v^-)$ $(v \in V(G'))$. We call $\mathcal{F}$ *integral* if all $\mathbb{P}_v^+$, $\mathbb{P}_v^-$ are integral. By an (*integral*) *chain* in $\mathcal{F}$ we mean any vector in $\mathbb{R}^{E(G')}$ $(\mathbb{Z}^{E(G')})$. A chain f in $\mathcal{F}$ is said to be a *flow* in $\mathcal{F}$ if

$$\begin{aligned} f(\Delta_v^+) = f(\Delta_v^-) \quad & \text{for any } v \in V(G'),\ v \neq s, t, \\ \sigma_v^+(X) \leq f(X) \leq \rho_v^+(X) \quad & \text{for any } v \in V(G') \text{ and } X \subseteq \Delta_v^+, \\ \sigma_v^-(X) \leq f(X) \leq \rho_v^-(X) \quad & \text{for any } v \in V(G') \text{ and } X \subseteq \Delta_v^-. \end{aligned}$$

If f is a flow in $\mathcal{F}$, then $v_f = f(\Delta_s^-) - f(\Delta_s^+)$ is called the *value* of f. Moreover,

if $U \subseteq V(G')$, then $f|(\Delta_U^+ \cup \Delta_U^-)$ is called a *U-transversal* of $\mathcal{F}$. The set of all U-transversals of $\mathcal{F}$ is called a *U-gammoid* of $\mathcal{F}$.

In the proof of the next theorem we show that this model can be expressed in the framework of flows in abstract networks. But in order to formulate this theorem we need another class of polyhedra from [K6].

If S_1 and S_2 are disjoint finite sets and $\mathbb{P} = (S_1 \cup S_2, \rho, \sigma)$ is a g-polymatroid, then $\mathbb{Q} = \{\mathbf{u} \oplus \mathbf{v}; \mathbf{u} \oplus -\mathbf{v} \in \mathbb{P}, \mathbf{u} \in \mathbb{R}^{S_1}, \mathbf{v} \in \mathbb{R}^{S_2}\}$ is called a *quasi polymatroid* (*q-polymatroid*) on the ordered couple of *ground sets* (S_1, S_2). Formally, we write $\mathbb{Q} = (S_1, S_2, \rho, \sigma)$ and $\mathbb{P}$ is called the *underlying* g-polymatroid of $\mathbb{Q}$. $\mathbb{Q}$ is called *integral* if $\mathbb{P}$ is integral. More details are in section 8.

THEOREM 6. *Let $\mathcal{F}$ be an (integral) g-polymatroidal flow network on a digraph G' with a source s, a sink t, and a collection of g-polymatroids $\mathbb{P}_v^+ = (\Delta_v^+, \rho_v^+, \sigma_v^+)$, $\mathbb{P}_v^- = (\Delta_v^-, \rho_v^-, \sigma_v^-)$ $(v \in V(G'))$. Suppose $\mathbb{P}_t^+ = \mathbb{R}^{\Delta_t^+}$, $\mathbb{P}_t^- = \mathbb{R}^{\Delta_t^-}$, and $\mathcal{F}$ admits a flow. Let $G' \setminus t$ denote the digraph obtained from G' after removing t and all its neighboring arcs. Then the t-gammoid of $\mathcal{F}$ is an (integral) q-polymatroid $\mathbb{Q} = (\Delta_t^+, \Delta_t^-, \rho, \sigma)$ where*

$$\begin{aligned}
\rho(X) &= \max\left\{f(X \cap \Delta_t^+) - f(X \cap \Delta_t^-);\ f \text{ is an (integral) flow in } \mathcal{F}\right\} \\
&= \min_{\substack{U \subseteq V(G') \\ s \in U, t \notin U}} \min_{Y \subseteq E(G' \setminus t)} \Bigg(\sum_{v \in U} -\sigma_v^+(\Delta_v^+ \cap (X \cup Y)) + \rho_v^-(\Delta_v^- \cap (X \cup Y)) \\
&\qquad + \sum_{v \in V(G') \setminus (U \cup t)} \rho_v^+(\Delta_v^+ \setminus (X \cup Y)) - \sigma_v^-(\Delta_v^- \setminus (X \cup Y)) \Bigg), \\
\sigma(X) &= \min\left\{f(X \cap \Delta_t^+) - f(X \cap \Delta_t^-);\ f \text{ is an (integral) flow in } \mathcal{F}\right\} \\
&= \max_{\substack{U \subseteq V(G') \\ s \in U, t \notin U}} \min_{Y \subseteq E(G' \setminus t)} \Bigg(\sum_{v \in U} -\rho_v^+(\Delta_v^+ \cap (X \cup Y)) + \sigma_v^-(\Delta_v^- \cap (X \cup Y)) \\
&\qquad + \sum_{v \in V(G') \setminus (U \cup t)} \sigma_v^+(\Delta_v^+ \setminus (X \cup Y)) - \rho_v^-(\Delta_v^- \setminus (X \cup Y)) \Bigg)
\end{aligned}$$

for any $X \subseteq \Delta_t$. Furthermore, if $\mathcal{F}$ is integral, then any integral t-transversal of $\mathcal{F}$ can be extended into an integral flow in $\mathcal{F}$.

Proof. For any $v \in V(G')$, let $\widetilde{\mathbb{P}}_v^-$ be the g-polymatroid on $-\Delta_v^-$ isomorphic with $\mathbb{P}_v^-$ under the isomorphism $x \mapsto -x$. Take an abstract network $\mathcal{N}_3$ on the underlying graph G of G' so that $\mathbb{P}_v = (\Delta_v, \rho_v, \sigma_v) = \mathbb{P}_v^+ \ominus \widetilde{\mathbb{P}}_v^-$ for any $v \in V(G) \setminus \{s, t\}$ (see Example 6) and $\mathbb{P}_s = \mathbb{P}_s^+ \oplus -\mathbb{P}_s^-$, $\mathbb{P}_t = \mathbb{P}_t^+ \oplus \mathbb{P}_t^-$. Let $\mathbb{Q}$ and $\mathbb{P}$ be the t-gammoids of $\mathcal{F}$ and $\mathcal{N}_3$, respectively. Then $\mathbf{u} \in \mathbb{Q}$ iff $(\mathbf{u}|\Delta_t^+) \oplus (-\mathbf{u}|\Delta_t^-) \in \mathbb{P}$. Therefore, by Theorem 5, $\mathbb{Q}$ is a q-polymatroid $(\Delta_t^+, \Delta_t^-, \rho, \sigma)$ so that for any $X \subseteq \Delta_t^+ \cup \Delta_t^-$ (see also (11)),

$$\begin{aligned}
\rho(X) &= \min_{Z \subseteq E(G \setminus t)} \sum_{v \in V(G \setminus t)} -\sigma_v(\Delta_v \cap (X \cup D(Z))) \\
&= \min_{Y \subseteq E(G' \setminus t)} \Bigg(-\sigma_s^+(\Delta_s^+ \cap (X \cup Y)) + \rho_s^-(\Delta_s^- \cap (X \cup Y)) \\
&\quad + \sum_{v \in V(G') \setminus \{s,t\}} \min\left\{-\sigma_v^+(\Delta_v^+ \cap (X \cup Y)) + \rho_v^-(\Delta_v^- \cap (X \cup Y)\right),
\end{aligned}$$

$$\left. \rho_v^+(\Delta_v^+ \setminus (X \cup Y)) - \sigma_v^-(\Delta_v^- \setminus (X \cup Y))\} \right)$$

$$= \min_{Y \subseteq E(G' \setminus t)} \min_{\substack{U \subseteq V(G') \\ s \in U, t \notin U}} \left(\sum_{v \in U} -\sigma_v^+(\Delta_v^+ \cap (X \cup Y)) + \rho_v^-(\Delta_v^- \cap (X \cup Y)) \right.$$

$$\left. + \sum_{v \in V(G') \setminus (U \cup t)} \rho_v^+(\Delta_v^+ \setminus (X \cup Y)) - \sigma_v^-(\Delta_v^- \setminus (X \cup Y)) \right).$$

The rest of the proof is either trivial or follows directly from Theorem 5. □

Note that if G' has an oriented loop e, then we can subdivide it by a new vertex v_e and take $\mathbb{P}_{v_e}^+$ and $\mathbb{P}_{v_e}^-$ to be the free g-polymatroids. Since $\mathbb{P}_{v_e}^+ \ominus \widetilde{\mathbb{P}}_{v_e}^-$ is the principal g-polymatroid, we get that Theorem 6 holds also if G' has oriented loops.

We shall apply this result for networks where $\Delta_t^- = \emptyset$. Then $\mathbb{Q}$ becomes a g-polymatroid $(\Delta_t^+, \rho, \sigma)$ so that ρ and σ satisfy the formulas from Theorem 6.

If we have a g-polymatroidal flow network $\mathcal{F}$ where all $\mathbb{P}_v^+$ and $\mathbb{P}_v^-$ are polymatroids, we get a *polymatroidal flow network* introduced by Lawler and Martel [LM1]. Now we can simplify Theorem 6.

THEOREM 7. *Let $\mathcal{F}$ be an (integral) polymatroidal flow network on a digraph G' with a source s, a sink t, and a collection of polymatroids $\mathbb{P}_v^+ = (\Delta_v^+, \rho_v^+)$, $\mathbb{P}_v^- = (\Delta_v^-, \rho_v^-)$ $(v \in V(G'))$. Let $\mathbb{P}_t^+$ be the free polymatroid on Δ_t^+ and $\Delta_t^- = \emptyset$. Then, the t-gammoid of $\mathcal{F}$ is an (integral) polymatroid $\mathbb{P} = (\Delta_t^+, \rho)$ such that*

$$\rho(X) = \max\{f(X);\ f \text{ is an (integral) flow in } \mathcal{F}\}$$

$$= \min_{\substack{U \subseteq V(G') \\ s \in U, t \notin U}} \min_{Z \subseteq \Delta_U^-} \left(\sum_{v \in U} \rho_v^-(\Delta_v^- \cap (X \cup Z)) \right.$$

$$\left. + \sum_{v \in V(G') \setminus (U \cup t)} \rho_v^+(\Delta_v^+ \cap (\Delta_U^- \setminus Z)) \right)$$

for any $X \subseteq \Delta_t^+$. Furthermore, if $\mathcal{F}$ is integral, then any integral t-transversal of $\mathcal{F}$ can be extended into an integral flow in $\mathcal{F}$.

Proof. Now $\sigma_v^+ \equiv 0$ and $\sigma_v^- \equiv 0$ for any $v \in V$. Thus, by Theorem 6,

$$\rho(X) = \min_{\substack{U \subseteq V(G') \\ s \in U, t \notin U}} \min_{Y \subseteq E(G' \setminus t)} \left(\sum_{v \in U} \rho_v^-(\Delta_v^- \cap (X \cup Y)) \right. \tag{12}$$

$$\left. + \sum_{v \in V(G') \setminus (U \cup t)} \rho_v^+(\Delta_v^+ \setminus (X \cup Y)) \right).$$

Let S and T be the sets of arcs from G' with both ends in U and $-U$ $(= V(G') \setminus U)$, respectively. If $u \in U$ and $w \in -U$, then from the monotonicity of ρ_w^+ and ρ_u^- we get

$$\rho_u^-(\Delta_u^- \cap (X \cup Y)) \geq \rho_u^-(\Delta_u^- \cap (X \cup (Y \setminus S))),$$
$$\rho_w^+(\Delta_w^+ \setminus (X \cup Y)) \geq \rho_w^+(\Delta_w^+ \setminus (X \cup Y \cup T)).$$

Thus the minimum in (12) occurs if $T \subseteq Y \subseteq E(G') \setminus S$, i.e., Y can be changed only on the set $\Delta_U^+ \cup \Delta_U^-$. Furthermore, let $R = Y \cap \Delta_U^+$. Then,

$$\Delta_u^- \cap (X \cup Y) = \Delta_u^- \cap (X \cup (Y \setminus R)),$$
$$\Delta_w^+ \setminus (X \cup Y) = \Delta_w^+ \setminus (X \cup (Y \setminus R)),$$

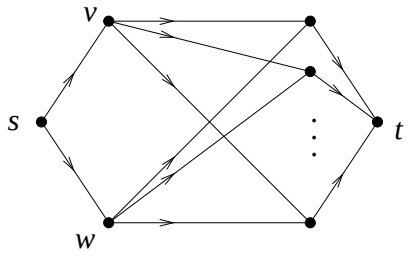


FIG. 3.

and the minimum in (12) occurs if $Y \cap \Delta_U^+ = \emptyset$, i.e., $T \subseteq Y \subseteq E(G') \setminus (S \cup \Delta_U^+)$. Therefore, we can take $Z \subseteq \Delta_U^-$ and transform (12) into the formula described in Theorem 7. □

6. Operations on polymatroids. Theorems 5, 6, and 7 can be used as general frameworks for describing almost all operations on matroids, polymatroids, and g-polymatroids. We give two transparent examples.

Example 7. Take the operation *sum of g-polymatroids* (see Frank and Tardos [FT]). Let $\mathbb{P}_1 = (S, \rho_1, \sigma_1)$ and $\mathbb{P}_2 = (S, \rho_2, \sigma_2)$ be two g-polymatroids. Then the sum of $\mathbb{P}_1$ and $\mathbb{P}_2$ is the g-polymatroid $\mathbb{P}_1 + \mathbb{P}_2 = (S, \rho_1 + \rho_2, \sigma_1 + \sigma_2)$ and $\mathbf{u} \in \mathbb{P}_1 + \mathbb{P}_2$ iff there exists $\mathbf{u}_1 \in \mathbb{P}_1$ and $\mathbf{u}_2 \in \mathbb{P}_2$ so that $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$ (i.e., $\mathbf{u}(s) = \mathbf{u}_1(s) + \mathbf{u}_2(s)$ for any $s \in S$). Furthermore, if $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbf{u}$ are integral, then $\mathbf{u}_1$ and $\mathbf{u}_2$ can be chosen to be integral.

Clearly, $\mathbb{P}_1 + \mathbb{P}_2$ is the t-gammoid of the g-polymatroidal flow network $\mathcal{F}_1$ on the digraph from Fig. 3 provided $\mathbb{P}_v^-$ is isomorphic with $\mathbb{P}_1$, $\mathbb{P}_w^-$ is isomorphic with $\mathbb{P}_2$ and, in all other cases, $\mathbb{P}_u^+$ ($\mathbb{P}_u^-$) is a free g-polymatroid on Δ_u^+ (Δ_u^-). The properties of $\mathbb{P}_1 + \mathbb{P}_2$ are consequences of Theorem 6. For instance, by Theorem 6 and equation (4), for any $X \subseteq S$,

$$\begin{aligned}\rho(X) &= \max\{f(X);\ f \text{ is an (integral) feasible flow in } \mathcal{F}_1\}\\ &= \rho_1(X) + \rho_2(X).\end{aligned}$$

Example 8. Now we describe the operation on polymatroids called $\mathbf{c}$-*dual*. It was introduced by McDiarmid [McD] (see also [We], [F2]). Let $\mathbb{P} = (S, \rho)$ be a polymatroid and $\mathbf{c} \in \mathbb{R}^{\mathbf{S}}$ such that $\mathbf{c}(\mathbf{X}) \geq \rho(\mathbf{X})$ for any $X \subseteq S$. Then the $\mathbf{c}$-dual of $\mathbb{P}$ is the polymatroid $\mathbb{P}^{\mathbf{c}} = (S, \rho^{\mathbf{c}})$, where $\rho^{\mathbf{c}}(X) = \mathbf{c}(\mathbf{X}) - \rho(\mathbf{S}) + \rho(\mathbf{S} \setminus \mathbf{X})$ for any $X \subseteq S$. Furthermore, $\mathbf{u} \in \mathbb{P}^{\mathbf{c}}$ iff there exists $\mathbf{y} \in \mathbb{P}$ such that $\mathbf{y}(\mathbf{S}) = \rho(\mathbf{S})$and $\mathbf{y} \leq \mathbf{c} - \mathbf{u}$ (i.e., $\mathbf{y}(\mathbf{s}) \leq \mathbf{c}(\mathbf{s}) - \mathbf{u}(\mathbf{s})$ for any $s \in S$).

Then $\mathbb{P}^{\mathbf{c}}$ is the t-gammoid of the g-polymatroidal flow network $\mathcal{F}_2$ on the digraph from Fig. 4 having the following properties: $\mathbb{P}_v^+$ is isomorphic with $\mathbb{P}$, e has the upper and the lower capacities equal to $\rho(S)$ (i.e., $\mathbb{P}_v^- = \mathbb{P}_s^+ = \{\rho(S)\}$), $\mathbf{c} = (\mathbf{c_1}, \ldots, \mathbf{c_n})$, and e_i' has the upper and lower capacities c_i and 0, respectively (i.e., $\mathbb{P}_{\mathbf{u}_i}^+ = (\Delta_{\mathbf{u}_i}^+, \rho_{\mathbf{u}_i}^+)$ satisfies $\rho_{\mathbf{u}_i}^+(\Delta_{\mathbf{u}_i}^+) = c_i$, $i = 1, \ldots, n$). All other constrained polymatroids are free. Note that only e has nonzero lower capacity and, therefore, the flow network is g-polymatroidal and not polymatroidal. But the t-gammoid is, in fact, a polymatroid. By Theorem 6,

$$\rho^{\mathbf{c}}(X) = \max\{f(X);\ f \text{ is an (integral) feasible flow in } \mathcal{F}_2\}.$$

Let $X = \{e_{i_1}, \ldots, e_{i_k}\}$. Using the greedy algorithm (see, e.g., [We]) we can arrange a flow g in $\mathcal{F}_2$ so that $g(\{e_{i_1}'', \ldots, e_{i_k}''\}) = \rho(S) - \rho(S \setminus X)$ and $g(\{e_{i_1}', \ldots, e_{i_k}'\}) = \mathbf{c}(\mathbf{X})$.

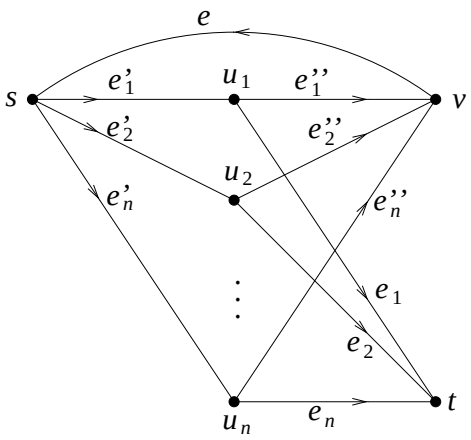


FIG. 4.

Then $g(\{e_{i_1},\dots,e_{i_k}\}) = \mathbf{c}(\mathbf{X}) - \rho(\mathbf{S}) + \rho(\mathbf{S}\setminus\mathbf{X})$. Furthermore, for any flow f in $\mathcal{F}_2$, $f(\{e''_{i_1},\dots,e''_{i_k}\}) \geq \rho(S) - \rho(S\setminus X)$ and $f(\{e_{i_1},\dots,e_{i_k}\}) \leq \mathbf{c}(\mathbf{X}) - \mathbf{f}(\{\mathbf{e''_{i_1}},\dots,\mathbf{e''_{i_k}}\}) \leq \mathbf{c}(\mathbf{X}) - \rho(\mathbf{S}) + \rho(\mathbf{S}\setminus\mathbf{X})$. Thus, $\rho^{\mathbf{c}}(X) = \mathbf{c}(\mathbf{X}) - \rho(\mathbf{S}) + \rho(\mathbf{S}\setminus\mathbf{X})$.

In Examples 7 and 8 we have described two operations using (4), (5), and Theorem 6. Similarly, other basic operations on matroids and (generalized) polymatroids can be described—for instance, sum, discrete sum, translation, truncation, (inverse) homomorphic image, intersections with a plank and a box. For definitions see Welsh [We], Aigner [A], Fujishige [F2], and Frank and Tardos [FT]. We have deleted a more detailed description because we believe that the above examples are sufficiently transparent and convincing.

7. Transversals, gammoids, and linking systems. In this section we show that Theorem 7 generalizes some results from transversal theory, especially the constructions of transversal matroids and gammoids. First we recall some known notions. By a *uniform matroid of rank* k we mean (S,ρ), where $\rho(X) = \min\{k, |X|\}$ for any $X \subseteq S$. If $k = |S|$, then we get the *free matroid* on S. Note that a matroid (S,ρ) is identified with the set $\{X \subseteq S; \rho(X) = |X|\}$ in this section (see, e.g., [O], [We]).

Let $\mathcal{A} = (A_i : i \in I)$ be a finite system of subsets of a finite set S. A subset X of S is called a *partial transversal* of $\mathcal{A}$ if there exists a bijection $\alpha : X \to I' \subseteq I$ such that $x \in A_{\alpha(x)}$ for any $x \in X$. By the theorem of Edmonds and Fulkerson [EF], the system of partial transversals of $\mathcal{A}$ form a matroid on S. Matroids of this kind are called *transversal matroids*. Clearly, the partial transversals of $\mathcal{A}$ are the t-transversals of the following polymatroidal flow network: Take a bipartite digraph with partition of vertices S, I such that if $v \in S$ and $i \in I$, then there exists a (v,i) arc iff $v \in A_i$. Add a source s, a sink t, the arcs (s,v) for every $v \in S$, and the arcs (i,t) for every $i \in I$. Let $\mathbb{P}^-_s$ be the free matroid on Δ^-_s, $\mathbb{P}^+_t$ be the free polymatroid on Δ^+_t, and all other constraints are uniform matroids of rank 1.

Gammoids generalize the transversal matroids and were introduced by Perfect [Pe] and Pym [Py] (see also Welsh [We] and Aigner [A]). They are constructed as follows: Take a digraph G'. For two subsets X, Y of $V(G')$ we say that X *can be linked into* Y if for some bijection $\varphi : X \to Y$ we can find vertex disjoint paths $(P_v : v \in X)$ in G' such that P_v has terminal vertex v and initial vertex $\varphi(v)$. Then,

for any digraph G' and any subsets S, T of $V(G')$, the set

$$L_S(G', T) = \{X \subseteq S; X \text{ can be linked into a } Y \subseteq T\}$$

is the collection of independent sets of a matroid on S. Matroids arising in this way are called *gammoids*. It is known that transversal matroids form a proper subclass of gammoids. See Welsh [We] or Aigner [A] for more details.

But $L_S(G', T)$ is isomorphic with the t-gammoid of a polymatroidal flow network $\mathcal{F}_3$ defined as follows: Let G'' be the digraph arising from G' after adding two new vertices s, t and the arcs directed from s to every $v \in S$ and from every $v \in T$ to t. Let s be the source and t be the sink of $\mathcal{F}_3$. Let $\mathbb{P}_s^-$ be the free matroid on Δ_s^-, $\mathbb{P}_t^+$ be the free polymatroid on Δ_t^+, and all other constraints are uniform matroids of rank 1. Then the t-gammoid of $\mathcal{F}_3$ is isomorphic with $L_S(G', T)$. Thus Theorem 7 generalizes the results of Edmonds and Fulkerson [EF], Perfect [Pe], and Pym [Py], and the notions of t-transversals and t-gammoids generalize the notions of transversals and gammoids, respectively.

In [K1], [K2], [K3], and [K5] we have generalized several results from transversal theory. Some of them can be covered by Theorem 7. For instance, [K5, Theorem 3] is, in fact, Theorem 7 restricted to polymatroidal flow networks on digraphs G' of the following type: The vertex set of G' consists of four sets $V_1 = s$, $V_2 = S$, $V_3 = T$, $V_4 = t$, and, furthermore, for every $i = 1, 2, 3$ and every $x \in V_i$, $y \in V_{i+1}$, there exists just one arc directed from x to y and there are no other arcs in G'. On the other hand it is an easy task to prove that Theorem 7 is equivalent with [K5, Theorem 3].

Another very interesting generalization of transversals and gammoids was presented by Schrijver [S2] using linking systems. A similar approach was introduced in Kung [Ku]. The *linking system* is a triple (S, T, Λ) where S and T are finite sets and Λ is a set of couples (X, Y) satisfying $|X| = |Y|$, $X \subseteq S$, $Y \subseteq T$, and other special conditions which we do not repeat here, because for us the fact (see [S2, Theorem 3.2]) that $\{Y \cup S \setminus X; (X, Y) \in \Lambda\}$ form a system of the basis of a matroid $(S \cup T, \rho_\Lambda)$ with a base S is more important (note that $B \subseteq S \cup T$ is a base if $\rho_\Lambda(S \cup T) = \rho_\Lambda(B) = |B|$). If (S, ρ) is a matroid and (S, T, Λ) a linking system, then the set

$$\{Y \subseteq T; (X, Y) \in \Lambda \text{ for a set } X \subseteq S \text{ satisfying } |X| = \rho(X)\}$$

is again a matroid (see [S2, Theorem 3.3]. But this matroid is the t-gammoid of the g-polymatroidal flow network $\mathcal{F}_4$ on the digraph from Fig. 5 so that $\mathbb{P}_v^- = (S \cup T, \rho_\Lambda)$, $\mathbb{P}_u^- = (S, \rho)$, $\mathbb{P}_v^+ = \{\rho_\Lambda(S \cup T)\}$, $\mathbb{P}_{u_i}^+$, $\mathbb{P}_{u_i}^-$, $\mathbb{P}_{v_j}^+$, $\mathbb{P}_{v_j}^-$ $(i = 1, \ldots, k,\ j = 1, \ldots, r)$ are all uniform matroids of rank 1 and all other constraints are free g-polymatroids. Thus Theorem 6 generalizes the constructions based on [S2, Theorem 3.3]. Note that the characteristic vectors of linking systems form a quasi polymatroid (to show this we can use a similar network as depicted in Fig. 5). Thus Theorem 3.3 from [S2] is also a direct corollary of Theorem 2. Analogously as polymatroids generalize matroids, polylinking systems generalize linking systems and have been introduced in Schrijver [S1] (see also the discussion in the last chapter).

Lawler and Martel [LM2] have used polymatroidal flow networks for formulating many problems from combinatorial optimization. In sections 6 and 7 we have shown that Theorems 5, 6, and 7 can play a similar role.

8. Properties of U-gammoids. If $\mathbb{Q} = (S_1, S_2, \rho, \sigma)$ is a q-polymatroid equal to a g-polymatroid on the ground set $S_1 \cup S_2$, then $\mathbb{Q}$ is called *improper*. Otherwise, it is called *proper*. For instance, if $\mathbb{P}_1$ and $\mathbb{P}_2$ are g-polymatroids on S_1 and S_2

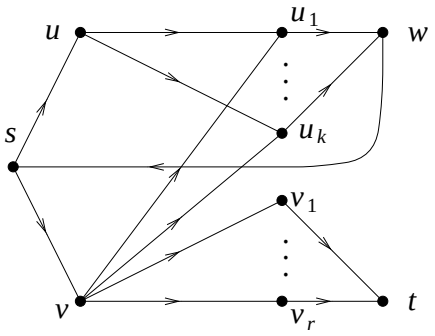


FIG. 5.

respectively, then $\mathbb{P}_1 \oplus \mathbb{P}_2$ is an improper q-polymatroid on (S_1, S_2) with the underlying g-polymatroid $\mathbb{P}_1 \oplus (-\mathbb{P}_2)$. Furthermore, if $S_2 = \emptyset$, then $\mathbb{Q} = \mathbb{P}_1$, i.e., every g-polymatroid is a q-polymatroid.

If $\overline{\mathbb{P}} = (\{a, b\}, \overline{\rho}, \overline{\sigma})$ is the principal g-polymatroid on $\{a, b\}$, then $\overline{\mathbb{Q}} = (a, b, \overline{\rho}, \overline{\sigma})$ is called the *principal q-polymatroid* on (a, b). By Example 2, $\overline{\mathbb{Q}} = \{\mathbf{u} \in \mathbb{R}^{\{a,b\}}; \mathbf{u}(a) = \mathbf{u}(b)\}$.

LEMMA 1. *The principal q-polymatroid $\overline{\mathbb{Q}}$ on $\{a, b\}$ satisfies:*

(a) *$\overline{\mathbb{Q}}$ is a proper q-polymatroid.*

(b) *$\overline{\mathbb{Q}}$ cannot be obtained as an intersection of two g-polymatroids on $\{a, b\}$.*

Proof. If a g-polymatroid $\mathbb{P} = (\{a, b\}, \rho, \sigma)$ contains $\overline{\mathbb{Q}}$, then, by (4), (5), and Example 2, $\mathbb{P} = \mathbb{R}^{\{a,b\}}$. This fact implies the statement. □

As pointed out in section 6, all basic operations on g-polymatroids can be described in the framework of Theorem 6, i.e., for any operation there exists a g-polymatroidal flow network $\mathcal{F}'$ with a source s, a sink t, $\Delta_t^- = \emptyset$, and $\mathbb{P}_t^+ = \mathbb{R}^{\Delta_t^+}$ so that the t-gammoid of $\mathcal{F}'$ is the resulting polyhedron of the operation.

On the other hand, any network $\mathcal{F}$ from Theorem 6 describes an operation on the g-polymatroids $\mathbb{P}_v^+$, $\mathbb{P}_v^-$ $(v \in V(G) \setminus t)$ so that the result of the operation is the t-gammoid of $\mathcal{F}$. But if $\Delta_t^+ \neq \emptyset \neq \Delta_t^-$, then the t-gammoid is a q-polymatroid. It is natural to ask whether it is a proper q-polymatroid. We deal with this problem in the next statement. A *weak circuit* C' in a digraph G' is any orientation of a circuit C from the underlying graph G of G', i.e., $E(C') \cup (-E(C')) = D(C)$. A circuit in G is any connected subgraph with all vertices of degree two.

PROPOSITION 1. *Let G' be a digraph with a source s and a sink t. Then there exists a g-polymatroidal flow network $\mathcal{F}$ on G' with $\mathbb{P}_t^+ = \mathbb{R}^{\Delta_t^+}$, $\mathbb{P}_t^- = \mathbb{R}^{\Delta_t^-}$, so that the t-gammoid of $\mathcal{F}$ is a proper q-polymatroid iff G' contains a weak circuit C' having an arc from Δ_t^+ and an arc from Δ_t^-.*

Proof. Necessity. Let G' have a weak circuit C' with the above property. Suppose $G' = C'$. Then choose $\mathcal{F}$ as follows: If $v \in V(G')$ and $|\Delta_v^+| = |\Delta_v^-| = 1$, then $\mathbb{P}_v^+$ and $\mathbb{P}_v^-$ are free g-polymatroids, and if $|\Delta_v^+| = 2$ $(|\Delta_v^-| = 2)$ then $\mathbb{P}_v^+$ $(\mathbb{P}_v^-)$ is the principal g-polymatroid on Δ_v^+ (Δ_v^-). Then the t-gammoid of $\mathcal{F}$ is the principal q-polymatroid; thus, by Lemma 1, it is proper.

If $G' \neq C'$, then take $\mathcal{F}$ such that the arcs not contained in C have capacity zero and the arcs from C are constrained as in the previous case. Then the t-gammoid of $\mathcal{F}$ is a direct sum of the principal q-polymatroid and the zero vector; thus it is a proper q-polymatroid (because its restriction is a proper q-polymatroid).

Sufficiency. Suppose G' does not contain a cycle C' with the above property.

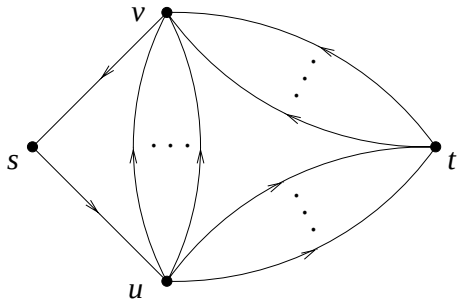


FIG. 6.

Then G' must contain two subdigraphs, G'_1 and G'_2, such that $V(G'_1) \cap V(G'_2) = t$, $E(G'_1) \cap E(G'_2) = \emptyset$, $E(G'_1) \cup E(G'_2) = E(G')$, $\Delta_t^+ \subseteq E(G'_1)$ and $\Delta_t^- \subseteq E(G'_2)$. If $\mathcal{F}$ is a g-polymatroidal flow network on G', then take $\mathcal{F}_1$ and $\mathcal{F}_2$ to be its restrictions to G'_1 and G'_2, respectively. Then the t-gammoids of $\mathcal{F}_1$ and $\mathcal{F}_2$ are g-polymatroids on Δ_t^+ and Δ_t^-, respectively. Thus the t-gammoid of $\mathcal{F}$ is an improper q-polymatroid (it is direct sum of the t-gammoids of $\mathcal{F}_1$ and $\mathcal{F}_2$). □

In the next example we show that there are very natural and simple operations on g-polymatroids with the resulting polyhedron equal to a q-polymatroid.

Example 9. Let S, T, R be finite pairwise disjoint sets and $\mathbb{P}_1 = (S \cup R, \rho_1, \sigma_1)$, $\mathbb{P}_2 = (T \cup R, \rho_2, \sigma_2)$ be (integral) g-polymatroids. Suppose $\rho_1(Z) \geq \sigma_2(Z)$, $\rho_2(Z) \geq \sigma_1(Z)$ for any $Z \subseteq R$. Then there exists an (integral) q-polymatroid $\mathbb{Q} = (S, T, \rho, \sigma)$ such that for any $X \subseteq S$, $Y \subseteq T$,

$$\rho(X \cup Y) = \min_{Z \subseteq R} \left(\rho_1(X \cup Z) - \sigma_2(Y \cup Z)\right),$$
$$\sigma(X \cup Y) = \max_{Z \subseteq R} \left(\sigma_1(X \cup Z) - \rho_2(Y \cup Z)\right).$$

Vector $\mathbf{u} \in \mathbb{R}^{S \cup T}$ ($\mathbf{u} \in \mathbb{Z}^{S \cup T}$) is from $\mathbb{Q}$ iff there exists a vector $\mathbf{v} \in \mathbb{R}^R$ ($\mathbf{v} \in \mathbb{Z}^R$) such that $(\mathbf{u}|S) \oplus \mathbf{v} \in \mathbb{P}_1$ and $(\mathbf{u}|T) \oplus \mathbf{v} \in \mathbb{P}_2$. $\mathbb{Q}$ is the t-gammoid of the g-polymatroidal flow network $\mathcal{F}_5$ on the digraph from Fig. 6, where $\mathbb{P}_u^- = \mathbb{P}_1$, $\mathbb{P}_v^+ = \mathbb{P}_2$, $\mathbb{P}_v^- = \mathbb{P}_s^+ = \mathbb{R}^{\Delta_s^+}$, and $\mathbb{P}_u^+ = \mathbb{P}_s^- = \mathbb{R}^{\Delta_s^-}$. The formulas for ρ and σ follows from Theorem 6. If $R = \emptyset$, then $\mathbb{Q} = \mathbb{P}_1 \oplus \mathbb{P}_2$. If $R \neq \emptyset$, then, by Proposition 1, $\mathbb{Q}$ can be a proper q-polymatroid. For instance, if R, S, T are singletons and $\mathbb{P}_1$, $\mathbb{P}_2$ are principal g-polymatroids, then $\mathbb{Q}$ is the principal q-polymatroid on (S, T).

From the results of Nakamura [N1], [N2], it follows that any q-polymatroid is a *universal polymatroid* (or, equivalently, the greedy algorithm always works on it). But the opposite implication does not hold. For instance, we can check that the convex hull of $\{\pm\mathbf{e_1}, \pm\mathbf{e_2}\}$ is a universal polymatroid in $\mathbb{R}^2$ but no q-polymatroid (see [N1] for more details).

Note that by [GLS] there exists a polynomial algorithm that finds the maximum of any objective function over the polytope arising as an intersection of finite number of q-polymatroids (or universal polymatroids). But to find an optimal integral vector from an intersection of a g-polymatroid and a q-polymatroid is NP-hard. This follows from results of Chandrasekaran and Kabadi [CK]. On the other hand if $\mathbb{Q}_1$ and $\mathbb{Q}_2$ are two integral q-polymatroids on the same couple of ground sets (S_1, S_2), then this problem has a polynomial algorithm because the following statement immediately follows from Theorem 1.

COROLLARY 3. *Let $\mathbb{Q}_1 = (S_1, S_2, \rho_1, \sigma_1)$ and $\mathbb{Q}_2 = (S_1, S_2, \rho_2, \sigma_2)$ be two q-polymatroids. Then the following linear system is totally dual integral:*

$$\sigma_i(X \cup Y) \leq \mathbf{u}(X) - \mathbf{u}(Y) \leq \rho_i(X \cup Y), \quad (i = 1, 2,\ X \subseteq S_1,\ Y \subseteq S_2).$$

Also, now, if $\mathbb{Q}_1$ and $\mathbb{Q}_2$ are integral q-polymatroids on (S_1, S_2), then $\mathbb{Q}_1 \cap \mathbb{Q}_2$ is an integral polyhedron.

Suppose $\mathcal{N}$ is an abstract network on a graph G, $U \subseteq V(G)$ and $W = V(G) \setminus U$. Let G_U (G_W) be the graph obtained from G after contracting the set U (W) into one new vertex u (w) and deleting the loops incident with u (w). Take $\mathcal{N}_U$ ($\mathcal{N}_W$) to be the abstract network on G_U (G_W) so that $\mathbb{P}_u = \mathbb{R}^{\Delta_u}$ ($\mathbb{P}_w = \mathbb{R}^{\Delta_w}$) and $\mathbb{P}_v$ be the same as in $\mathcal{N}$ for any $v \in W$ ($v \in U$). Let $\mathbb{P}_U$ ($\mathbb{P}_W$) be the u-gammoid of $\mathcal{N}_U$ (w-gammoid of $\mathcal{N}_W$). By Theorem 5, $\mathbb{P}_U$ and $\mathbb{P}_W$ are g-polymatroids. Note that $\Delta_u = \Delta_U = -\Delta_W = -\Delta_w$. Let $\overline{\mathbb{P}}_W$ be the g-polymatroid isomorphic with $-\mathbb{P}_W$ under the isomorphism $x \mapsto -x$. Then the U-gammoid of $\mathcal{N}$ is the intersection of $\mathbb{P}_U$ and $\overline{\mathbb{P}}_W$. Thus we can conclude.

THEOREM 8. *Let $\mathcal{N}$ be an (integral) abstract network on a graph G, $U \subseteq V(G)$ and $W = V(G) \setminus U$. Then the U-gammoid of $\mathcal{N}$ is the (integral) polyhedron arising as intersection of the (integral) g-polymatroids $\mathbb{P}_U$ and $\overline{\mathbb{P}}_W$. Furthermore, if $\mathcal{N}$ is integral, then any integral U-transversal can be extended into an integral flow from $\mathcal{N}$.*

In the proof of Theorem 6 we have shown that g-polymatroidal flow networks are special cases of abstract networks. Then, from Proposition 1 and Corollary 3 follows the next statement.

PROPOSITION 2. *If $\mathcal{F}$ is an (integral) g-polymatroidal flow network on a digraph G' and $U \subseteq V(G')$, then the U-gammoid of $\mathcal{F}$ can be obtained as an intersection of two (integral) q-polymatroids on (Δ_U^+, Δ_U^-). Furthermore, if $\mathcal{F}$ is integral, then any integral U-transversal can be extended into an integral flow from $\mathcal{F}$.*

Note that using Lemma 1 and the ideas from the proof of Proposition 1 we can show that if G' contains a weak circuit C' having arcs from Δ_U^+ and Δ_U^-, then there exists a g-polymatroidal flow network on G' so that its U-gammoid cannot be obtained as an intersection of two g-polymatroids.

Clearly, the classical flow model is just a special case of g-polymatroidal flow networks. Therefore, Proposition 2 gives information about the behavior of flows on edge cuts in the classical model, too.

9. Concluding remarks. By Fujishige [F1] (see also [FT], [F2], [S3]), g-polymatroids are the projections of base polyhedra on a basis hyperplane. Therefore, g-polymatroids are not a substantial generalization of polymatroids and also Theorem 1 is equivalent to the Edmonds intersection theorem. Similarly, the flow model presented in section 4 is equivalent not only to the models from [K6], [LM3] but also to the model from [LM1].

But g-polymatroids are the most suitable polyhedra for expressing Theorem 2. Define, for example, an operation *partial intersection* of polyhedra $\mathbb{P}_1$ in $\mathbb{R}^{S \cup T}$ and $\mathbb{P}_2$ in $\mathbb{R}^T$ to be a polyhedron $\mathbb{P} = \{\mathbf{u} \in \mathbb{R}^S; \mathbf{u} \oplus \mathbf{v} \in \mathbb{P}_1 \text{ for a } \mathbf{v} \in \mathbb{P}_2\}$. Then Theorem 2, in fact, says that the class of (integral) g-polymatroids is closed under the operation of partial intersection. Furthermore, this operation unifies and generalizes other operations and constructions as have been shown in the paper. Such a simple formulation cannot be used if we deal with polymatroids (if $\mathbb{P}_1$ and $\mathbb{P}_2$ are polymatroids, then $\mathbb{P} = \mathbb{P}_1|S$, and this cannot play such an universal role as Theorem 2). If we want to have an analogous universal tool for constructing polymatroids, we must

deal with polylinking systems of Schrijver [S1], which form a class of polyhedra formally different from the class of polymatroids. But in Theorem 2, we deal only with g-polymatroids.

Similarly, the flow model from section 4 is the most suitable model for formulating Theorems 5 and 8. For instance, formulation of Theorem 6 and Proposition 2 are more clumsy because we must use q-polymatroids.

Acknowledgment. The author expresses thanks to S. Poljak and the unknown referees for many valuable comments.

REFERENCES

[A] M. Aigner, *Combinatorial Theory*, Springer-Verlag, Berlin, 1979.

[CK] R. Chandrasekaran and S. N. Kabadi, *Pseudomatroids*, Discrete Math., 71 (1988), pp. 205–217.

[DMcD] J. Davies and C. McDiarmid, *Disjoint common transversals and exchange structures*, J. London Math. Soc., 14 (1976), pp. 55–62.

[E] J. Edmonds, *Submodular functions, matroids and certain polyhedra*, in Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Shönheim, eds., Gordon and Breach, New York, 1970, pp. 69–87.

[EF] J. Edmonds and D. R. Fulkerson, *Transversals and matroid partition*, J. Res. Nat. Bur. Standards Sect. B, 69B (1965), pp. 147–153.

[EG] J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, in Studies in Integer Programming in Ann. Discrete Math., Vol. 1, P. L. Hammer, E. L. Johnson, and B. H. Korte, eds., North–Holland, Amsterdam, 1977, pp. 185–204.

[FF] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[Fr] A. Frank, *Generalized polymatroids*, in Finite and Infinite Sets, A. Hajnal and L. Lovász, eds., North–Holland, Amsterdam, 1984, pp. 285–294.

[FT] A. Frank and É. Tardos, *Generalized polymatroids and submodular flows*, Math. Programming, 42 (1988), pp. 489–563.

[F1] S. Fujishige, *A note on Frank's generalized polymatroids*, Discrete Appl. Math., 7 (1984), pp. 105–109.

[F2] S. Fujishige, *Submodular Functions and Optimization*, Ann. Discrete Math., Vol. 47, North–Holland, Amsterdam, 1991.

[GLS] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[Ha] R. Hassin, *Minimum cost flow with set-constraints*, Networks, 12 (1982), pp. 1–21.

[Ho] A. J. Hoffman, *A generalization of max-flow min-cut*, Math. Programming, 6 (1974), pp. 352–359.

[J] F. Jaeger, *Nowhere-zero flow problems*, in Selected Topics in Graph Theory 3, L. W. Beineke and R. J. Wilson, eds., Academic Press, New York, 1988, pp. 71–95.

[KC] S. N. Kabadi and R. Chandrasekaran, *On totally dual integral systems*, Discrete Appl. Math., 26 (1990), pp. 87–104.

[K1] M. Kochol, *Some Generalizations of Transversal Theory*, CSc. thesis, Slovak Academy of Sciences, Bratislava, Slovakia, 1990.

[K2] M. Kochol, *The notion and basic properties of M-transversals*, Discrete Math., 104 (1992), pp. 191–196.

[K3] M. Kochol, *About a generalization of transversals*, Math. Bohem., 119 (1994), pp. 143–149.

[K4] M. Kochol, *Compatible systems of representatives*, Discrete Math., 132 (1994), pp. 115–126.

[K5] M. Kochol, *Symmetrized and continuous generalization of transversals*, Math. Bohem., 121 (1996), pp. 95–106.

[K6] M. Kochol, *Quasi polymatroidal flow networks*, Acta Math. Univ. Comenian., 64 (1995), pp. 83–97.

[K7] M. Kochol, *Superposition and constructions of graphs without nowhere-zero k-flows*, manuscript.

[K8] M. Kochol, *Hypothetical complexity of the nowhere-zero 5-flow problem*, J. Graph Theory, 28 (1998), pp. 1–11.

[Kov] M. M. Kovalev, *Maximization of convex functions on supermatroids*, Dokl. Akad. Nauk BSSR, 27 (1983), pp. 584–587 (in Russian).

[KP] M. M. Kovalev and N. N. Pisaruk, *Generalized matroids*, Dokl. Akad. Nauk BSSR, 28 (1984), pp. 972–975 (in Russian).

[Ku] J. P. S. Kung, *Bimatroids and invariants*, Adv. Math., 30 (1978), pp. 238–249.

[LM1] E. L. Lawler and C. U. Martel, *Computing maximal "polymatroidal" network flows*, Math. Oper. Res., 7 (1982), pp. 334–347.

[LM2] E. L. Lawler and C. U. Martel, *Flow network formulation of polymatroid optimization problem*, in Bonn Workshop on Combinatorial Optimization, A. Bachem, M. Grötschel, and B. Korte, eds., Ann. Discrete Math., Vol. 16, North–Holland, Amsterdam, 1982, pp. 189–200.

[LM3] E. L. Lawler and C. U. Martel, *Polymatroidal flows with lower bounds*, Discrete Appl. Math., 15 (1986), pp. 291–313.

[L1] L. Lovász, *Flats in matroids and geometric graphs*, in Combinatorial Surveys, P. Cameron, ed., Academic Press, New York, 1977, pp. 45–86.

[L2] L. Lovász, *Submodular functions and convexity*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 235–257.

[McD] C. J. H. McDiarmid, *Rado's theorem for polymatroids*, Math. Proc. Cambridge Philos. Soc., 78 (1975), pp. 263–281.

[MP] L. Mirsky and H. Perfect, *Application of the notion of independence to combinatorial analysis*, J. Combin. Theory, 2 (1968), pp. 327–357.

[M1] K. Murota, *Valuated matroid intersection,* I*: Optimality criteria*, SIAM J. Discrete Math., 9 (1996), pp. 545–561.

[M2] K. Murota, *Valuated matroid intersection,* II*: Algorithms*, SIAM J. Discrete Math., 9 (1996), pp. 562–576.

[M3] K. Murota, *Convexity and Steinitz's exchange property*, Adv. Math., 124 (1996), pp. 272–311.

[N1] M. Nakamura, *A characterization of greedy sets: Universal polymatroids* (I), Sci. Papers College Arts Sci. Univ. Tokyo, 38 (1988), pp. 155–167.

[N2] M. Nakamura, *An intersection theorem for universal polymatroids: Universal polymatroids* (II), Sci. Papers College Arts Sci. Univ. Tokyo, 40 (1990), pp. 95–100.

[N3] M. Nakamura, *Δ-polymatroids and an extension of Edmonds-Giles' TDI scheme*, in Proc. IPCO 93, Erice, Italy, 1993, pp. 401–412.

[NW] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley, New York, 1988.

[O] J. Oxley, *Matroid Theory*, Oxford University Press, London, 1992.

[Pe] H. Perfect, *Applications of Menger's graph theorem*, J. Math. Anal. Appl., 22 (1968), pp. 96–111.

[Pu] W. R. Pulleyblank, *Polyhedral combinatorics*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 235–257.

[Py] J. S. Pym, *A proof of the linkage theorem*, J. Math. Anal. Appl., 27 (1969), pp. 636–639.

[R] A. Recski, *Matroid Theory and Its Applications in Electric Network Theory and Statics*, Springer-Verlag, Berlin, 1989.

[S1] A. Schrijver, *Matroids and Linking Systems*, Mathematical Centre Tracts, Vol. 88, Mathematisch Centrum, Amsterdam, 1978.

[S2] A. Schrijver, *Matroids and linking systems*, J. Combin. Theory Ser. B, 26 (1979), pp. 349–369.

[S3] A. Schrijver, *Total dual integrality from directed graphs, crossing families, and sub- and supermodular functions*, in Progress in Combinatorial Optimization W. R. Pulleyblank, ed., Academic Press, Toronto, Ontario, 1984, pp. 315–362.

[S4] A. Schrijver, *Proving total dual integrality with cross-free families - A general framework*, Math. Programming, 29 (1984), pp. 15–27.

[S5] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley, New York, 1986.

[T] E. Tardos, *An intersection theorem for supermatroids*, J. Combin. Theory Ser. B, 50 (1990), pp. 150–159.

[We] D. J. A. Welsh, *Matroid Theory*, Academic Press, London, 1976.

[Wo] D. R. Woodall, *Vector transversals*, J. Combin. Theory Ser. B, 32 (1982), pp. 189–205.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 487–500, August 1998

010

CONGESTION-FREE ROUTINGS OF LINEAR COMPLEMENT PERMUTATIONS*

MARK RAMRAS†

Abstract. We present an off-line method for routing a linear complement permutation on a hypercube. The routing has the virtue of being congestion-free. Our method is purely algebraic, and the routing involves row reducing an invertible matrix to the identity by means of special row operations.

Key words. hypercube, routing, congestion-free, linear complement permutation

AMS subject classifications. 05C, 68

PII. S0895480196301461

1. Introduction.

In a previous paper [7] we obtained congestion-free routings for bit permute complement (BPC) permutations on a hypercube. Routings for such permutations were also studied by Z. Liu and J.-H. You [3], and by D. Nassimi and S. Sahni [4, 5]. Using the conceptual framework of our earlier paper [6], we obtain congestion-free routings for any linear complement (LC) permutation on a hypercube. An LC permutation is one in which an n-bit string (expressed as a column vector) is multiplied on the left by an invertible $n \times n$ matrix over GF(2), and then certain specified bits of the resulting column vector are complemented. Alternatively, an LC permutation amounts to an affine map (a linear map followed by a translation) of the n-cube $Q_n \simeq \mathcal{Z}_2^n$. Since a bit permute (BP) permutation π is a linear permutation whose associated matrix M is the permutation matrix obtained by permuting the rows of the identity matrix, every BPC permutation is an LC permutation. Algorithms for routing LC permutations have been given by Boppana and Raghavendra [1] and by Zemoudeh and Sengupta [9]. However, these routings are not congestion-free. At certain stages of the routing, some nodes will contain two messages, while others have none. F. T. Leighton [2, Problem 3.191, p. 758] considers the routing of linear permutations and suggests a method somewhat similar to the one we present here.

2. Preliminaries.

By Q_n we mean the n-dimensional hypercube. π will denote a permutation of $V(Q_n)$, the nodes of Q_n. By $d(x, y)$ we mean the (Hamming) distance in Q_n between nodes x and y. By the *weight* of x we mean the number of 1's in the n-tuple x, i.e., $d(x, 0)$. For any subset B of $\{1, 2, \ldots, n\}$, the *complementation* σ_B is the permutation of Q_n defined by $\sigma_B(x) = x + \sum_{i \in B} e_i$, where $\{e_1, \ldots, e_n\}$ denotes the standard basis of $\mathcal{Z}_2^n = Q_n$.

We now recall some definitions from [6].

[6, Definition 1.4]. $k(\pi) = \max\{d(x, \pi(x)) | x \in V(Q_n)\}$.
$\Delta = \{\pi \in Perm(Q_n) | k(\pi) = 1\}$.

[6, Definition 1.1]. $t_\Delta(\pi) = \min\{t | \pi \in \Delta^t\}$, where Δ^t is the set of all t-fold products of elements of Δ.

As explained in the introduction of [6], a representation of π as an element of Δ^t can naturally be identified with a t-step congestion-free routing of π, where by

*Received by the editors April 3, 1996; accepted for publication (in revised form) November 5, 1997; published electronically July 7, 1998.
http://www.siam.org/journals/sidma/11-3/30146.html
†Department of Mathematics, Northeastern University, Boston, MA 02115 (mbramras@neu.edu).

congestion-free we mean that at no time does any node contain more than one message. Thus, $t_\Delta(\pi)$ is the minimum number of steps in a congestion-free routing of π. Clearly, *any* routing of π requires at least $k(\pi)$ steps (in a single step a message can stay put or else travel a distance of 1), and so $t_\Delta(\pi) \geq k(\pi)$. The purpose of this paper is to show that for π any linear complement permutation of the nodes of Q_n, congestion-free routings are easy to construct. Moreover, each step of such a routing is again a linear complement permutation.

A few words about notation are necessary. We shall express permutations as products of cycles and denote by $(1, 2, \ldots, m)$ the cycle which maps i to $i+1$ for $1 \leq i \leq m-1$, and m to 1. We multiply cycles from right to left so that cycle multiplication behaves exactly like composition of functions.

3. LU decompositions and routings.

DEFINITION 3.1. *For any invertible $n \times n$ matrix M with entries in $GF(2)$, let π_M be the permutation of the nodes of Q_n defined by $\pi_M(x) = M \cdot x$ (thinking of x as an $n \times 1$ column vector).*

LEMMA 3.2. *Let A be an $n \times n$ matrix such that $I + A$ is invertible. Suppose that for all $x \in Q_n$, weight$(A \cdot x) \leq 1$. Then A has at most one nonzero row. If that row is the ith, then $A_{ii} = 0$. Conversely, if A has at most one nonzero row, then for all $x \in Q_n$, weight$(A \cdot x) \leq 1$.*

Proof. Suppose that A_i, the ith row of A, is not zero. Suppose that $A_{ik} \neq 0$ and, for some j and l with $j \neq i$, $A_{jl} \neq 0$. If $k = l$, then *weight*$(A \cdot e_k) \geq 2$. If $k \neq l$, then *weight*$(A \cdot (e_k + e_l)) \geq 2$, so no other row of A is nonzero.

Next, suppose that $A_{ii} \neq 0$. Then $(I + A)_{ii} = 0$. Therefore, the ith row of $I + A$ is a sum of some subset of the *other* rows of A. Hence the rows of $I + A$ are linearly dependent, contradicting the assumed invertibility of $I + A$.

Now, for the converse assume that the ith row of A is nonzero, and all other rows are zero. For any $x \in Q_n$, for $j \neq i$, the jth component of the column vector $A \cdot x$ is zero. Thus *weight*$(A \cdot x) \leq 1$. □

COROLLARY 3.3. *If $M \neq I$ and M is invertible, then $\pi_M \in \Delta \Leftrightarrow I + M$ has exactly one nonzero row, and if that row is the ith, then $M_{ii} = 1$. Conversely, if $I + M$ has row i as its unique nonzero row, and if $M_{ii} = 1$, then M is invertible.*

Proof. Let $A = I + M$. Then $M = I + A$, and the result follows from Lemma 3.2. For the last assertion, note that by the hypotheses, M and I agree in every row except (possibly) the ith, and $M_{ii} = 1$. Computing $\det(M)$ by expanding along rows, it is easy to see that $\det(M) = 1$, and so M is invertible. □

COROLLARY 3.4. *If $\pi_M \in \Delta$, then $M^{-1} = M$.*

Proof. We may assume that $M \neq I$. Let $A = I + M$. Then since $I + A = M$, by Lemma 3.2 A has exactly one nonzero row, say the ith, and $A_{ii} = 0$. Hence $A^2 = 0$, and so

$$M^2 = (I + A)^2 = I + A^2 = I.$$

Hence $M^{-1} = M$. □

LEMMA 3.5. *$k(\pi_M) = n \Leftrightarrow$ the vector $[1, 1, \ldots, 1]^T \in$ range$(I + M)$.*

Proof. $k(\pi_M) = n \Leftrightarrow$ for some $x \in Q_n$, $d(x, \pi_M(x)) = n$. Now

$$d(x, \pi_M(x)) = \textit{weight}(x + M \cdot x) = \textit{weight}((I + M)x),$$

and since the only vector of weight n is $[1, 1, \ldots, 1]^T$, the result follows. □

COROLLARY 3.6. *If both M and $I + M$ are invertible, then $k(\pi_M) = n$.*

Proof. If $(I + M)$ is invertible, then *range*$(I + M) = Q_n$, so the result follows from Lemma 3.5. ∎

LEMMA 3.7. *Let A be a nonzero $n \times n$ matrix. Then weight$(e_i + A \cdot x) \leq 1$ for all $x \Leftrightarrow A$ has at most two nonzero rows, one of which is the ith, and if there is a second nonzero row, it is equal to the ith.*

Proof. ($\Rightarrow$) First suppose that row i of A is zero, and suppose that row j is nonzero. Say $A_{jk} \neq 0$. Then $A \cdot e_k$ is the kth column of A and therefore has a 1 in the jth row. It also must have a 0 in the ith row. Hence *weight*$(e_i + A \cdot e_k) \geq 2$, contradicting the hypothesis. So row i is nonzero. Next we will show that if row j is also nonzero, then it is equal to row i. The ith component of $e_i + A \cdot x$ is $1 + A_i \cdot x$, where A_i denotes the ith row of A, and the jth component is $A_j \cdot x$. Since $e_i + A \cdot x$ has weight ≤ 1, $A_i \cdot x = 0 \Rightarrow A_j \cdot x = 0$. Thus,

$$\textit{nullspace}(A_i) \subseteq \textit{nullspace}(A_j),$$

and since both subspaces have dimension $n - 1$, they are equal. Hence so are their orthogonal complements. However, these are just $\{0, A_i\}$ and $\{0, A_j\}$. Therefore, A_j and A_i must be equal. Thus, any two nonzero rows of A must be equal. Lastly, we must show that no *more* than two rows of A can be nonzero. So suppose that $A_j = A_l = A_i \neq 0$. Then for some q, $A_{jq} = A_{lq} = A_{iq} \neq 0$. Let $x = e_q$. Then $A \cdot x$ is the qth column of A, which has at least three 1's. Hence $e_i + A \cdot e_q$ has at least two 1's, contradicting the hypothesis. Therefore, A has at most two nonzero rows.

($\Leftarrow$) If A_i is the only nonzero row of A, then for all x, $A \cdot x$ is either 0 or e_i. Hence $e_i + A \cdot x$ is either e_i or 0. On the other hand, if $A_j = A_i$ and all other rows of A are zero, then if $\alpha = A_i \cdot x$, the jth entry of $e_i + A \cdot x$ is α and the ith is $1 + \alpha$. Since exactly one of these is nonzero, and all other entries are zero, *weight*$(e_i + A \cdot x) = 1$. ∎

COROLLARY 3.8. *If $B \neq \emptyset$ and $M \neq I$, then $\sigma_B \pi_M \in \Delta \Leftrightarrow B = \{i\}$, for some i, and $I + M$ has at most two nonzero rows, one of which is the ith, and if $I + M$ has a second nonzero row, it is equal to the ith.*

Proof. ($\Leftarrow$) $\sigma_{\{i\}}\pi_M(x) = e_i + M \cdot x$, so $d(x, \sigma_{\{i\}}\pi_M(x)) = \textit{weight}(e_i + (I + M)(x))$. Let $A = I + M$. Then $d(x, \sigma_{\{i\}}\pi_M(x)) = \textit{weight}(e_i + A \cdot x)$. Since A satisfies the conditions of Lemma 3.7, $k(\sigma_{\{i\}}\pi_M) = 1$.

($\Rightarrow$) It suffices, by Lemma 3.7, to show that $|B| = 1$. Since $\sigma_B \pi_M \in \Delta$, $d(0, \sigma_B \pi_M(0)) = 1$, i.e., *weight* $(\sum_{j \in B} e_B) = 1$. Hence $|B| = 1$. ∎

Remark. Suppose that ϕ is an LC permutation and $\phi \in \Delta$. Let M be an invertible $n \times n$ matrix such that $M \neq I$. If $\phi = \pi_M$, then ϕ moves messages along edges of only one dimension. If $\phi = \sigma_{\{i\}}\pi_M$, then ϕ moves messages along edges of at most two different dimensions.

Suppose that $\phi = \pi_M$ and that the unique nonzero row of $I + M$ is the ith. Then if x is adjacent to $\phi(x)$, the edge $\langle x, \phi(x)\rangle$ has the same dimension as the edge $\langle 0, x + \phi(x)\rangle = \langle 0, (I + M) \cdot x\rangle$, which is i. Now suppose that $\phi = \sigma_{\{i\}}\pi_M$. By Corollary 3.8, $I + M$ has precisely two nonzero rows, which are equal and one of which is the ith. If the other is the jth, then the nonzero components of any edge $\langle x, \phi(x)\rangle$ are the ith and the jth. Hence these are the dimensions of the edges along which ϕ moves messages.

We should point out that as with *any* member of Δ, $k(\phi) = 1$, i.e., for all x, $d(x, \phi(x)) = 1$. So for each message x, x and $\phi(x)$ are adjacent, and thus ϕ moves x only along the single edge $\langle x, \phi(x)\rangle$.

LEMMA 3.9. *Suppose that A is an $n \times n$ matrix whose two nonzero rows are equal. Let $M = I + A$. The following are equivalent:*

(1) *M is invertible;*
(2) *$A^2 = 0$;*
(3) *$M^2 = I$;*
(4) *$M^{-1} = M$.*

Proof. (1) $\Rightarrow$ (2) If $A_i = A_j = R \neq 0$ are the two nonzero rows of A, then $(A^2)_i = (A^2)_j = R \cdot A$ are the only two (possibly) nonzero rows of A^2. Now $R \cdot A = (R_i + R_j)R$, where R_k denotes the kth entry of R. So $A^2 = (R_i + R_j)A$. Now the scalar $R_i + R_j$ is either 0 or 1, so A^2 is either A or 0. If $A^2 = A$, then $A(I + A) = 0$, and since $I + A$ is assumed to be invertible, $A = 0$, contrary to our hypothesis. Hence $A^2 = 0$.

(2) $\Rightarrow$ (3) If $A^2 = 0$, then $M^2 = (I + A)^2 = I + A^2 = I$.

(3) $\Rightarrow$ (4) and (4) $\Rightarrow$ (1) are both obvious. □

We now state a result whose proof is contained in the proof of (1) $\Rightarrow$ (2) of the preceding lemma.

LEMMA 3.10. *Let A be an $n \times n$ matrix whose only nonzero rows are rows i and j, which are both equal to $R = [R_1, \ldots, R_n]$. Then $A^2 = 0 \Leftrightarrow R_i = R_j$.*

COROLLARY 3.11. *Let A be as in Lemma 3.10 and $M = I + A$. Assume that $A_{ii} = A_{ij}$. Then*

(i) *M is invertible.*
(ii) *If $A_{ii} = 0$, then $(\sigma_{\{i\}}\pi_M)^{-1} = \sigma_{\{i\}}\pi_M$.*
(iii) *If $A_{ii} = 1$, then $(\sigma_{\{i\}}\pi_M)^{-1} = \sigma_{\{j\}}\pi_M$.*

Proof. (i) follows immediately from Lemmas 3.9 and 3.10.

(ii) Suppose that $A_{ii} = 0$. Then $A \cdot e_i = i$th column of $A = 0$. Hence $M \cdot e_i = (I + A) \cdot e_i = e_i$. So

$$\sigma_{\{i\}}\pi_M(\sigma_{\{i\}}\pi_M(x)) = \sigma_{\{i\}}\pi_M(e_i + M \cdot x) = e_i + M \cdot e_i + M^2 \cdot x.$$

By Lemma 3.9, $M^2 = I$, so

$$(\sigma_{\{i\}}\pi_M)^2(x) = e_i + M \cdot e_i + x = e_i + e_i + x.$$

Thus, $(\sigma_{\{i\}}\pi_M)^2 = I$.

(iii) Now suppose that $A_{ii} = 1$. Then $A \cdot e_i = e_i + e_j$. Hence $M \cdot e_i = e_i + (e_i + e_j) = e_j$. So

$$\sigma_{\{j\}}\pi_M(\sigma_{\{i\}}\pi_M)(x) = e_j + M \cdot e_i + x = e_j + e_j + x = x.$$

Hence $(\sigma_{\{j\}}\pi_M)(\sigma_{\{i\}}\pi_M) = I$. □

Combining Corollaries 3.4, 3.8, and 3.11 we have the following.

COROLLARY 3.12. (1) $\phi \in \Delta \Rightarrow \phi^{-1} \in \Delta$.

(2) *For any k, $\phi \in \Delta^k \Rightarrow \phi^{-1} \in \Delta^k$.*

Proof. (1) Let $\phi \in \Delta$. If $\phi = \pi_M$, then $M^{-1} = M$ and so $(\pi_M)^{-1} = \pi_{M^{-1}} = \pi_M$, and thus $\phi^{-1} \in \Delta$. If $\phi = \sigma_{\{i\}}\pi_M$, then by Corollary 3.11, ϕ^{-1} is either $\sigma_{\{i\}}\pi_M$ or $\sigma_{\{j\}}\pi_M$. Thus, by Corollary 3.8, $\phi^{-1} \in \Delta$.

(2) Let $\phi = \phi_1\phi_2 \cdots \phi_k$, where each $\phi_i \in \Delta$. Then

$$\phi^{-1} = (\phi_k)^{-1} \cdots (\phi_2)^{-1}(\phi_1)^{-1}.$$

By (1), each $(\phi_i)^{-1} \in \Delta$, and so $\phi \in \Delta^k$. □

Examples. (1) Let $M = \left[\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{smallmatrix}\right]$. Then $\pi_M(\left[\begin{smallmatrix} 1 \\ 0 \\ 1 \end{smallmatrix}\right]) = \left[\begin{smallmatrix} 0 \\ 1 \\ 0 \end{smallmatrix}\right]$, so $k(\pi_m) = 3$.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so $\pi_M \in \Delta^3$.

(2) Let $M = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{smallmatrix}\right]$. Then $A = I + M = \left[\begin{smallmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right]$, so for all x, *weight*$(A \cdot x) \leq 2$. Since *weight*$(A \cdot e_3) = 2$, $k(\pi_M) = 2$.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then $\pi_M = \pi_{M_2}\pi_{M_1}$, so $\pi_M \in \Delta^2$.

(3) Let $M = \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{smallmatrix}\right]$. Then $A = I + M = \left[\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{smallmatrix}\right]$, so for all x, *weight*$(A \cdot x) \leq 2$. Since *weight* $(A \cdot e_2) = 2$, $k(\pi_M) = 2$.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = M_2 \cdot M_1.$$

Now $\sigma_{\{2\}}\pi_{M_1}$, $\sigma_{\{3\}}\pi_{M_2} \in \Delta$, and $\pi_{M_2} \cdot \sigma_{\{2\}} = \sigma_{\{3\}} \cdot \pi_{M_2}$. Hence

$$\pi_M = (\sigma_{\{3\}}\pi_{M_2}) \cdot (\sigma_{\{2\}}\pi_{M_1}) \in \Delta^2.$$

PROPOSITION 3.13. *If the invertible $n \times n$ matrix M has an LU decomposition, then $M_{11} = 1$. Furthermore, if $M = LU$, then $M^{-1} = U^{-1}L^{-1}$ and $(M^{-1})_{nn} = 1$.*

Proof. First note that since M is invertible, so are L and U, and thus each has only 1's on its main diagonal. Since L is lower triangular, L_1, the first row of L is e_1. Since U is upper triangular, U^1, the first column of U, is e_1^T, the transpose of e_1. Hence $M_{11} = L_1 \cdot U^1 = e_1 \cdot e_1^T = 1$. For the second statement, the inverse of an upper triangular matrix with 1's on the main diagonal has the same form, and the corresponding statement is true for a lower triangular matrix with 1's on the main diagonal. Hence $(M^{-1})_{nn} = (U^{-1})_n \cdot (L^{-1})_n = e_n \cdot e_n^T = 1$. ∎

DEFINITION 3.14. *For any $n \times n$ matrix M, let $M_{\{i\}}$ denote the $(i-1)$ by $(i-1)$ upper left minor of M, i.e., the $(i-1) \times (i-1)$ submatrix obtained from M by deleting row k and column k for all $k \geq i$.*

PROPOSITION 3.15. *If $M = LU$, then $M_{\{n\}} = L_{\{n\}}U_{\{n\}}$.*

Proof. Note that $L_{\{n\}}$ is lower triangular, $U_{\{n\}}$ is upper triangular, and both have 1's on their main diagonals. Let $1 \leq i, j \leq n-1$. Then for $i < k \leq n$, $L_{ik} = 0$, and for $j < k \leq n$, $U_{kj} = 0$. Hence

$$(M_{\{n\}})_{ij} = M_{ij} = L_i \cdot U_j = \sum_{k \leq \min\{i,j\}} L_{ik}U_{kj} = (L_{\{k\}}U_{\{k\}}). \quad ∎$$

COROLLARY 3.16. *If $M = LU$, then for all $k \leq n$, $M_{\{k\}} = L_{\{k\}}U_{\{k\}}$.*

Proof. This follows by repeated application of Proposition 3.15, since

$$(M_{\{n\}})_{\{k\}} = M_{\{k\}}. \quad ∎$$

COROLLARY 3.17. *If $M = LU$, then there exists a $D \in \mathcal{D} = \{B|\pi_B \in \Delta\}$ such that the last row of DM is e_n. Furthermore, $D = I + A$, where the only nonzero row of A (if $A \neq 0$) is the nth and $A_{nn} = 0$. Hence D is lower triangular, as is DL, and DM has an LU decomposition.*

Proof. By Proposition 3.13, $(M^{-1})_{nn} = 1$. Since $I = M^{-1}M$,

$$e_n = (n\text{th row of } M^{-1})M = a_1M_1 + \cdots + a_{n-1}M_{n-1} + M_n.$$

Let $D = I + A$, where A is the matrix whose only nonzero row is the nth, which is equal to $(a_1, a_2, \ldots, a_{n-1}, 0)$. Then D is lower triangular, and hence so is DL. The last row of DM is e_n. Finally, $DM = (DL)U$ *is* the LU decomposition of DM. □

LEMMA 3.18. *Let $D = I + A$, where the only nonzero row of A is the kth, and suppose that $A_{kk} = 0$. Then π_D changes only the kth bit of an n-tuple. The same is true for $\sigma_{\{k\}}\pi_D$.*

Proof. Let x be an n-tuple and let $1 \leq i \leq n$, $i \neq k$. Then $\pi_D(x) = D \cdot x = x + A \cdot x$, where A_i is the ith row of A. Since $i \neq k$, $A_i = 0$, and so the ith component of $D \cdot x$ is x_i. This proves the first assertion. The second follows since $\sigma_{\{k\}}$, complementation of the kth bit, has the same property. □

We come now to the main result of this section.

THEOREM 3.19. *For $n \geq 2$, if the invertible $n \times n$ matrix M has an LU decomposition, then $M = D_nD_{n-1}\cdots D_1$, where D_i has all 1's on the main diagonal, and if D_i is not the identity matrix I, it differs from I only in the ith row. Hence $\pi_M \in \Delta^n$, i.e., there is an n-step routing for π_M. Moreover, in the ith step, only edges of dimension i are used.*

Proof. First suppose that $n = 2$. By Proposition 3.13, $M = \begin{bmatrix} 1 & x \\ y & z \end{bmatrix}$. Since M is invertible, x and Z cannot both be 0. If $x = 0$, then $z = 1$ and so $M = \begin{bmatrix} 1 & 0 \\ y & 1 \end{bmatrix} \in \mathcal{D}$ and thus $\pi_M \in \Delta \subseteq \Delta^2$. If $x = 1$ and $z = 0$, then $M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$. Let $D_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Then $D_2M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = D_1 \in \mathcal{D}$. Hence $M^{-1} = D_1D_2$, and so $M = D_2^{-1}D_1^{-1} = D_2D_1 \in \mathcal{D}^2$. Thus, $\pi_M = \pi_{D_2}\pi_{D_1} \in \Delta^2$. Finally, if $x = z = 1$, then $M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \in \mathcal{D}$, and so $\pi_M \in \Delta \subseteq \Delta^2$.

Now assume that $n \geq 3$. We shall show that there exist $n \times n$ matrices $D_1, D_2, \ldots, D_n$ such that for each i, $D_i \in \mathcal{D}$, D_i has only 1's on the main diagonal and differs from I, if at all, only in row i, and $D_1D_2\cdots D_nM = I$. We argue inductively. Since $M = LU$, it follows from Corollary 3.17 that there exists an $n \times n$ matrix $D_n \in \mathcal{D}$ such that D_n has all 1's on its main diagonal and differs from I (if at all) only in row n, and $M_1 = D_nM$ has last row $= e_n$. Now suppose that $D_{n-1}, D_{n-2}, \ldots, D_{n-k} \in \mathcal{D}$ have been found so that each D_{n-i} has only 1's on its main diagonal, differs from I only in row $n - i$, and so that the last k rows of $M_k = D_{n-k}\cdots D_nM$ agree with the last k rows of the identity matrix I. By Corollary 3.16, $(M_k)_{\{n-k\}}$ has an LU decomposition. Thus, by Corollary 3.17 there is an $(n-k-1) \times (n-k-1)$ matrix $D \in \mathcal{D}$ such that the last row of $D \cdot (M_k)_{\{n-k\}}$ is the last row of the $(n-k) \times (n-k)$ identity matrix. In other words, $0^{n-k-1}1$ is the sum of certain rows of $(M_k)_{\{n-k\}}$, one of which is the last row. We must show that $0^{n-k-1}10^k$, the $(n-k)$th row of the $n \times n$ identity matrix, can be expressed as the sum of the $(n-k)$th row of M_k and certain other rows of M_k. Let R_j be the jth row of $(M_k)_{\{n-k\}}$, and let S_j be the jth

row of M_k. Then for $1 \le j \le n-k$, the jth row of (M_k) is

$$S_j = R_j, x_{j1}, x_{j2}, \ldots, x_{jk} = R_j 0^k + \sum_{i=1}^{k} x_{ji} S_{n-k+i}.$$

Therefore, $R_j 0^k = S_j + \sum_{i=1}^{k} x_{ji} S_{n-k+i}$. We are assuming that the last k rows of M_k are the same as the last k rows of the identity matrix, and that $0^{n-k-1}1 = R_{n-k} + \sum_{j=1}^{n-k-1} \epsilon_j R_j$, where each $\epsilon_j \in \{0,1\}$. It follows that

$$0^{n-k-1}10^k = R_{n-k}0^k + \sum_{j=1}^{n-k-1} \epsilon_j R_j 0^k$$

$$= S_{n-k} + \sum_{i=1}^{k} x_{n-k,i} S_{n-k+i} + \sum_{j=1}^{n-k-1} \epsilon_j \left(S_j + \sum_{i=1}^{k} x_{ji} S_{n-k+i} \right).$$

Thus the $(n-k)$th row of the $n \times n$ identity matrix equals

$$S_{n-k} + \sum_{j=1}^{n-k-1} \epsilon_j S_j + \sum_{i=1}^{k} \left(x_{n-k,i} + \sum_{j=1}^{n-k-1} \epsilon_j x_{ji} \right) S_{n-k+i}.$$

In other words, the $(n-k)$th row of the identity matrix can be expressed as the sum of the $(n-k)$th row of M_k and certain other rows of M_k. Let $D_{n-k-1} \in \mathcal{D}$ be the matrix obtained from the identity matrix by replacing the $(n-k)$th row by $(\epsilon_1, \ldots, \epsilon_{n-k-1}, 1, \alpha_{n-k+1}, \ldots, \alpha_n)$, where

$$\alpha_{n-k+i} = x_{n-k,i} + \sum_{j=1}^{n-k-1} \epsilon_j x_{ji}.$$

Then $M_{k+1} = D_{n-k-1} M_k$ agrees with the identity matrix in its last $k-1$ rows. Thus, we have established the inductive step, and so the desired $D_1, D_2, \ldots, D_n$ all exist. It follows that $D_n D_{n-1} \cdots D_1 M = I$ and so $M = D_1 D_2 \cdots D_n$. Hence $\pi_M = \pi_{D_1} \pi_{D_2} \cdots \pi_{D_n} \in \Delta^n$. The final assertion is a consequence of Lemma 3.18. □

LEMMA 3.20. *Let D_i be an $n \times n$ matrix with 1's on the main diagonal and suppose that D_i differs from the identity matrix only in row i. Then $\sigma_{\{i\}} \pi_{D_i} = \pi_{D_i} \sigma_{\{i\}}$, and if $j \ne i$,*

$$\sigma_{\{j\}} \pi_{D_i} = \begin{cases} \pi_{D_i} \sigma_{\{j\}} & \text{if } (D_i)_{ij} = 0, \\ \sigma_{\{i\}} \pi_{D_i} \sigma_{\{j\}} & \text{if } (D_i)_{ij} = 1. \end{cases}$$

Proof. Since the ith column of D_i is e_i, the ith column of the identity matrix,

$$(\pi_{D_i} \sigma_{\{i\}})(x) = D_i \cdot (x + e_i) = D_i \cdot x + D_i \cdot e_i = D_i \cdot x + e_i = (\sigma_{\{i\}} \pi_{D_i})(x),$$

so $\sigma_{\{i\}} \pi_{D_i} = \pi_{D_i} \sigma_{\{i\}}$. Now assume that $j \ne i$. We have

$$(\pi_{D_i} \sigma_{\{j\}})(x) = D_i \cdot (x + e_j) = D_i \cdot x + D_i \cdot e_j = D_i \cdot x + j\text{th column of } D_i.$$

The jth column of D_i is either e_j or $e_j + e_i$, according to whether the (i,j)th entry of D_i is 0 or 1. Hence $\pi_{D_i}\sigma_{\{j\}}$ is either $\sigma_{\{j\}}\pi_{D_i}$ or $\sigma_{\{i\}}\sigma_{\{j\}}\pi_{D_i}$. If the latter, then by multiplying by$\sigma_{\{i\}}$ on the left we get $\sigma_{\{i\}}\pi_{D_i}\sigma_{\{j\}} = \sigma_{\{j\}}\pi_{D_i}$. □

THEOREM 3.21. *Let $S \subseteq \{1,2,\ldots,n\}$ and let σ_S denote the permutation which complements those bits which belong to S. Let M be any $n \times n$ matrix which has an LU decomposition. Then the LC permutation $\sigma_S \circ \pi_M$ belongs to Δ^n. The ith step in the n-step routing of $\sigma_S \circ \pi_M$ is either π_{D_i} or $\sigma_{\{i\}}\pi_{D_i}$, where $D_i \in \mathcal{D}$ differs from the identity matrix only in row i, and so only edges of dimension i are used during this step.*

Proof. If $S = \emptyset$, Theorem 3.19 applies. So assume $S \neq \emptyset$. By Theorem 3.19, $M = D_1 D_2 \cdots D_n$, where each D_i has 1's on the main diagonal and differs from I only in row i (if at all). Let $\varphi = (\sigma_{A_n}\pi_{D_n}) \cdots (\sigma_{A_1}\pi_{D_1})$, where each A_i is either $\emptyset$ or $\{i\}$. Note that by Lemma 3.18, the ith factor, $(\sigma_{A_i}\pi_{D_i})$, changes only the ith component of any n-tuple. Thus, such a factorization of φ expresses φ as an element of Δ^n, and provides a routing in which only edges of dimension i are used during step i. Now σ_S is the product (in any order) of those $\sigma_{\{i\}}$ for which $i \in S$. To prove the theorem, it suffices, therefore, to show that for any $k \in \{1,2,\ldots,n\}$, $\sigma_{\{k\}}\varphi$ has a factorization of the same form. However, by Lemma 3.20, for $i \neq k$, $\sigma_{\{k\}}(\sigma_{A_i}\pi_{D_i}) = \sigma_{A_i}(\sigma_{\{k\}}\pi_{D_i}) =$ either $\sigma_{A_i}(\sigma_{\{i\}}\pi_{D_i})\sigma_{\{k\}}$ or $(\sigma_{A_i}\pi_{D_i})\sigma_{\{k\}}$. Since $A_i \subseteq \{i\}$, $\sigma_{A_i} \cdot \sigma_{\{i\}} = \sigma_{B_i}$, where

$$B_i = \begin{cases} \{i\} & \text{if } A_i = \emptyset, \\ \emptyset & \text{if } A_i = \{i\}. \end{cases}$$

Thus, $\sigma_{\{k\}}(\sigma_{A_k}\pi_{D_k}) =$ either $(\sigma_{A_i}\pi_{D_i})\sigma_{\{k\}}$ or $(\sigma_{B_i}\pi_{D_i})\sigma_{\{k\}}$. Thus, we can keep moving $\sigma_{\{k\}}$ past each factor, replacing A_i by B_i until we get to $\sigma_{A_k}\pi_{D_k}$. Then $\sigma_{\{k\}}(\sigma_{A_k}\pi_{D_k}) = \sigma_{B_k}\pi_{D_k}$. Hence

$$\sigma_{\{k\}}\varphi = (\sigma_{B_n}\pi_{D_n}) \cdots (\sigma_{B_k}\pi_{D_{\{k\}}})(\sigma_{A_{k-1}}\pi_{D_{k-1}}) \cdots (\sigma_{A_1}\pi_{D_1}),$$

thereby proving the claim. □

4. Row permutations. Not every invertible matrix M has an LU decomposition. Sometimes it is necessary first to permute the rows of M. This is the case, for example, with permutation matrices, that is, matrices obtained from the identity matrix by a permutation of its rows. However, as is well known (see, for example, [8]), for *any* invertible M, there is a permutation matrix P such that $PM = LU$. Now there are some such matrices M for which π_M has a routing of the type discussed in Theorem 3.19.

Example 4.1. Let $M = \left[\begin{smallmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{smallmatrix}\right]$. Clearly M does not have an LU decomposition since $M_{11} = 0$. However, if P is the permutation matrix which interchanges rows 1 and 3, then PM does have an LU decomposition. Nevertheless, we can express M as an element of $\mathcal{D}^3$. To see this, we shall row-reduce M to the identity matrix by a sequence of three row operations, each of which corresponds to an element of $\mathcal{D}$.

$$M \to D_1 M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\to D_2 D_1 M = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\rightarrow D_3D_2D_1M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = I.$$

Example 4.2. Let $M = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$. Again, M has no LU decomposition. The following sequence of three row operations, each corresponding to an element of $\mathcal{D}$, reduces M to the identity.

$$(1)\ R_1 \leftarrow R_1 + R_2,\ (2)\ R_3 \leftarrow R_2 + R_3,\ (3)\ R_2 \leftarrow R_1 + R_2 + R_3.$$

Example 4.3. Let $M = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$. Then $M^{-1} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$. We claim that $\pi_{M^{-1}} \in \Delta^4$. The following sequence of row operations reduces M^{-1} to the identity.

$$(1)\ R_4 \leftarrow R_1 + R_2 + R_3 + R_4,\ (2)\ \sigma_{\{2\}} \circ \begin{pmatrix} R_2 \leftarrow R_3 + R_4 \\ R_3 \leftarrow R_2 + R_4 \end{pmatrix},$$

$$(3)\ \sigma_{\{2\}} \circ \begin{pmatrix} R_1 \leftarrow R_2 \\ R_2 \leftarrow R_1 \end{pmatrix},\ (4)\ R_2 \leftarrow R_2 + R_4$$

Note that by Corollary 3.8, steps (3) and (4) correspond to elements of Δ. Since each of the operations (1)−(4) is its own inverse, performing them in reverse order reduces M to the identity.

DEFINITION 4.4. *An $n \times n$ matrix C is of type* I *if it differs from the identity in only one row and has all* 1*'s on its main diagonal. C is of type* II *if it differs from the identity in exactly two rows, say rows i and j, $C_i + C_j = e_i + e_j$, and all entries on its main diagonal are* 1*'s. C is of type* III *if it differs from the identity in exactly two rows, and the matrix obtained by interchanging these two rows is of type* II.

LEMMA 4.5. *Any $n \times n$ matrix C of type* I, II, *or* III *is invertible.*

Proof. Suppose C is of type I and differs from I only in row i. Then since $C_{ii} = 1$ by hypothesis, it follows from Corollary 3.3 that C is invertible.

Next, suppose that C is of type II. Then $C_i + e_i = C_j + e_j$, and $C_{ii} = C_{jj} = 1$. Let $A = I + C$. Then $A_i = A_j$, and so $A_{ij} = A_{jj} = 1 + C_{jj} = 0$, while $A_{ii} = 1 + C_{ii} = 0$. Hence $A_{ii} = A_{ij}$. Thus, by Corollary 3.11, C is invertible.

Finally, if C is of type III, then $C = PC'$, where $P = P^{-1}$ is a permutation matrix and C' is a matrix of type II. So C is a product of invertible matrices and therefore is invertible. □

PROPOSITION 4.6. *Let M_0 be an $n \times n$ invertible matrix. There is a sequence of matrices $C_1, C_2, \ldots, C_n$ with each C_i of type* I *or* III*, such that $C_n \cdots C_2 C_1 M_0 = I$.*

Proof. Let R_1 be the first row of M_0^{-1}. Then $R_1 M_0 = e_1$. If $R_{11} = 1$, let C_1 be the matrix whose first row is R_1, and which agrees with I in all other rows. Thus, C_1 is of type I. If $R_{11} = 0$, let i_1 be the least i such that $R_{1i} = 1$. Let C_1 be the matrix whose first row is R_1, i_1th row is $R_1 + e_1 + e_i$, and which agrees with I in all other rows. Thus C_1 is of type III. In either case, $C_1 M_0 = M_1$ has first row equal to e_1.

Now assume that $k \geq 2$ and that for all $j < k$, C_j has been chosen and $C_j M_{j-1} = M_j$ agrees with I in its first j rows. We shall define C_k. Let R_k be the kth row of $(M_{k-1})^{-1}$. Then $R_k M_{k-1} = e_k$. Since M_{k-1} agrees with I in its first $k-1$ rows, the same is true for $(M_{k-1})^{-1}$. The rows of the latter matrix are linearly independent, and

thus so are the first k rows. Therefore, R_k is not a linear combination of $\{e_1, \ldots, e_{k-1}\}$. Hence for some $i \geq k$, $R_{ki} = 1$. Let i_k be the least such i. If $i_k = k$ (so that $R_{kk} = 1$), let C_k be the matrix whose kth row is R_k, and which agrees with I in all other rows. Then C_k is of type I. If $i_k > k$, let C_k be the matrix whose kth row is R_k, i_kth row is $R_k + e_k + e_{i_k}$, and which agrees with I in all other rows. Then C_k is of type III. In either case, $C_k M_{k-1} = M_k$ agrees with I in its first k rows.

So by induction, we obtain the desired sequence $C_1, \ldots, C_n$ and since M_n agrees with I in all n rows, $C_n C_{n-1} \cdots C_1 M_0 = M_n = I$. □

Note. If C is of type I, then $\pi_C \in \Delta$. Also, $\sigma_{\{i\}}\pi_C \in \Delta$, where the row of C which differs from I is the ith. On the other hand, if C is of type III, then $\pi_C \notin \Delta$. However, $\sigma_{\{i\}}\pi_C$, $\sigma_{\{j\}}\pi_C \in \Delta$, where the two rows of C which differ from I are rows i and j.

LEMMA 4.7. *Let C be an $n \times n$ matrix with 1's on the main diagonal, and suppose that C differs from I only in rows i and j. Assume further that $C_i + C_j = e_i + e_j$ and $C_{ij} = 0 = C_{ji}$. Let $F = PC$, where P is the permutation matrix which interchanges rows i and j. Let $k \neq i, j$. Then*

(1) $\pi_C \sigma_{\{i\}} = \sigma_{\{i\}} \pi_C$.
(2) $\pi_F \sigma_{\{i\}} = \sigma_{\{j\}} \pi_F$.
(3) *if* $C_{ik} = 0$, *then* $\pi_C \sigma_{\{k\}} = \sigma_{\{k\}} \pi_C$.
(4) *if* $C_{ik} = 1$, *then* $\pi_C \sigma_{\{k\}} = \sigma_{\{i,j,k\}} \pi_C$.
(5) *if* $F_{jk} = 0$, *then* $\pi_F \sigma_{\{k\}} = \sigma_{\{k\}} \pi_F$.
(6) *if* $F_{jk} = 1$, *then* $\pi_F \sigma_{\{k\}} = \sigma_{\{i,j,k\}} \pi_F$.

Proof. To establish the identity $A = B$, it suffices to prove that for an arbitrary column vector x, $Ax = Bx$.

(1) $\pi_C \sigma_{\{i\}} x = C(e_i + x) = Ce_i + Cx$. Now Ce_i is the ith column of C, which is e_i. Hence $\pi_C \sigma_{\{i\}} x = e_i + Cx = \sigma_{\{i\}} \pi_C x$.

(2) $\pi_F \sigma_{\{i\}} = \pi_P \pi_C \sigma_{\{i\}} = \pi_P(\sigma_{\{i\}} \pi_C)$, where the second equality follows from (1). But since P interchanges rows i and j, $\pi_P \sigma_{\{i\}} = \sigma_{\{j\}} \pi_P$. Hence $\pi_F \sigma_{\{i\}} = \sigma_{\{j\}} \pi_F$.

(3) and (4) $C_{ik} = C_{jk}$; by hypothesis, $C_{ik} + C_{jk} = e_{ik} + e_{jk}$ and, since $k \neq i, j$, $e_{ik} = e_{jk} = 0$.

For (3), assume that $C_{ik} = 0$. Then $C_{jk} = 0$, also. So kth column of $C = e_k$, and thus $Ce_k = e_k$. Therefore,

$$\pi_C \sigma_{\{k\}} x = C(e_k + x) = e_k + Cx = \sigma_{\{k\}} \pi_C x.$$

For (4), assume that $C_{ik} = 1$. Then $C_{jk} = 1$, also. If $q \notin \{i, j, k\}$, then $C_{qk} = 0$. Thus the kth column of C is $e_i + e_j + e_k$. Hence

$$\pi_C \sigma_{\{k\}} x = C(e_k + x) = e_i + e_j + e_k + Cx = \sigma_{\{i,j,k\}} \pi_C x.$$

(5) Assume $F_{jk} = 0$. Then $(PC)_{jk} = 0$. Since left multiplication by P interchanges rows i and j, $C_{ik} = 0$. Then by (3), we have $|pi_C \sigma_{\{k\}} = \sigma_{\{k\}} \pi_C$. Hence

$$\pi_F \sigma_{\{k\}} = \pi_P \pi_C \sigma_{\{k\}} = \pi_P \sigma_{\{k\}} \pi_C.$$

But since the kth column of P is e_k, $\pi_P \sigma_{\{k\}} = \sigma_{\{k\}} \pi_P$, and so $\pi_F \sigma_{\{k\}} = \sigma_{\{k\}} \pi_P \pi_C = \sigma_{\{k\}} \pi_F$.

(6) Assume $F_{jk} = 1$. Then $(PC)_{jk} = 1$. Now PC is the matrix obtained from C by interchanging rows i and j, so $C_{ik} = 1$. Hence by (4), $\pi_C \sigma_{\{k\}} = \sigma_{\{i,j,k\}} \pi_C$. So

$$\pi_F \sigma_{\{k\}} = \pi_P \pi_C \sigma_{\{k\}} = \pi_P \sigma_{\{i,j,k\}} \pi_C = \sigma_{\{i,j,k\}} \pi_P \pi_C = \sigma_{\{i,j,k\}} \pi_F. \quad \square$$

Note. The hypotheses of Lemma 4.7 are symmetric in i and j, so each of $(1)-(6)$ remains true when i and j are interchanged.

THEOREM 4.8. *Let M be an invertible $n\times n$ matrix, and let $\mathcal{J}(M)$ be the subset of $\{1,2,\ldots,n\}$ such that $j\in\mathcal{J}(M)\Leftrightarrow M$ differs from I in row j. Then $\pi_M\in\Delta^{|\mathcal{J}(M)|}$. In fact, if $S\subseteq\mathcal{J}(M)$, then $\sigma_S\pi_M\in\Delta^{|\mathcal{J}(M)|}$.*

Proof. The first assertion is a special case of the second since, when $S=\emptyset$, σ_S is the identity map. Let $\mathcal{J}=\mathcal{J}(M)$. We shall prove the second assertion by induction on $|\mathcal{J}|$. First suppose that $|\mathcal{J}|=1$, and suppose that $\mathcal{J}=\{j\}$. Then for $i\neq j$, row $M_i=e_i$. Since M is invertible, its rows are linearly independent, and so in particular, $M_j\notin span\{e_i|i\neq j\}$. Hence $M_{jj}=1$. Therefore, $\pi_M\in\Delta$. Also, by Corollary 3.8, $\sigma_{\{j\}}\pi_M\in\Delta$.

Now suppose that $|\mathcal{J}|\geq 2$ and that the result is true for all invertible matrices M' such that $|\mathcal{J}(M')|<|\mathcal{J}|$ and for all subsets S' of $\mathcal{J}'=\mathcal{J}(M')$. Let S be any subset of $\mathcal{J}$ and let k be any element of $\mathcal{J}$. Thus, $M_k\neq e_k$ and for $i\notin\mathcal{J}$, $M_i=e_i$. Let R be the kth row of M^{-1}. There are two cases according to whether $R_k=(M^{-1})_{kk}=1$ or 0.

Case 1. $(M^{-1})_{kk}=1$. Let C be the matrix whose kth row is R and which agrees with I in all other rows. Then since $C_{kk}=1$, all entries on the main diagonal of C are 1 and so by Corollaries 3.3, 3.4, and 3.8, both π_C and $\sigma_{\{k\}}\pi_C$ belong to Δ and $C^{-1}=C$. Let $M'=CM$. Then M' agrees with I in those rows in which M does, and also in row k. Hence $\mathcal{J}'=\mathcal{J}(M')=\mathcal{J}\setminus\{k\}$, and so $|\mathcal{J}'|=|\mathcal{J}(M)|-1$. Let $S'=S\cap\mathcal{J}'$, so that S is either S' or $S'\cup\{k\}$. By the induction hypothesis, $\sigma_{S'}\pi_{M'}\in\Delta^{|\mathcal{J}|-1}$. Now $M=C^{-1}M'=CM'$, so $\sigma_{S'}\pi_M=\sigma_{S'}\pi_C\pi_{M'}$. By Lemma 3.20, $\sigma_{S'}\pi_C$ is either $\pi_C\sigma_{S'}$ or $\sigma_{\{k\}}\pi_C\sigma_{S'}$ (according to whether the weight of row C_k is even or odd). So $\sigma_{S'}\pi_M$ is either $\pi_C\sigma_{S'}\pi_{M'}$ or $(\sigma_{\{k\}}\pi_C)\sigma_{S'}\pi_{M'}$. Hence $\sigma_{S'}\pi_M\in\Delta^{|\mathcal{J}|-1+1}=\Delta^{|\mathcal{J}|}$. On the other hand, $\sigma_S\pi_M=\sigma_{\{k\}}\sigma_{S'}\pi_M=\sigma_{\{k\}}\sigma_{S'}\pi_C\pi_{M'}$. Thus, $\sigma_S\pi_M$ is either $\sigma_{\{k\}}\pi_C\pi_{M'}$ or $\pi_C\pi_{S'}\pi_{M'}$. Since both π_C and $\sigma_{\{k\}}\pi_C$ belong to Δ, and $\pi_{S'}\pi_{M'}\in\Delta^{|\mathcal{J}|-1}$, in either case we have $\sigma_S\pi_M\in\Delta^{|\mathcal{J}|}$.

Case 2. We may now assume that for all $k\in\mathcal{J}$, $R_k=(M^{-1})_{kk}=0$. Choose any $k\in\mathcal{J}$ and any l such that $R_l=1$. Let C be the matrix whose kth row is R, whose lth row is $R+e_k+e_l$, and which agrees with I in all other rows. Then as in the proof of Proposition 4.6, C is of type III, and if $M'=CM$, then $M'_k=e_k$ and $\mathcal{J}'=\mathcal{J}(M')=\mathcal{J}\setminus\{k\}$. Let $S'=S\cap\mathcal{J}'$. Thus, by induction $\sigma_{S'}\pi_{M'}\in\Delta^{|\mathcal{J}|-1}$. By Lemma 3.9, $C^{-1}=C$. So $M=CM'$ and thus $\sigma_S\pi_M=\sigma_S\pi_C\pi_{M'}$. Suppose first that $S=\emptyset$. Then $\sigma_S=I$ and $\pi_M=\pi_C\pi'_M$. There are two possibilities for l: (i) $l\in\mathcal{J}(M')$ and (ii) $l\notin\mathcal{J}(M')$. Suppose that $l\in\mathcal{J}(M')$. Then by our induction hypothesis, $\sigma_{\{l\}}\pi_{M'}\in\Delta^{|\mathcal{J}'|}=\Delta^{|\mathcal{J}|-1}$. Now

$$\pi_C\pi_{M'}=(\sigma_{\{k\}}\sigma_{\{k\}})\pi_C\pi_{M'}=\sigma_{\{k\}}(\pi_C\sigma_{\{l\}})\pi_{M'}=(\sigma_{\{k\}}\pi_C)(\sigma_{\{l\}}\pi_{M'}),$$

and since $\sigma_{\{k\}}\pi_C\in\Delta$, it follows that $\pi_C\pi_{M'}\in\Delta^{1+|\mathcal{J}'|}=\Delta^{|\mathcal{J}|-1}$. So now suppose that $l\notin\mathcal{J}'$. Thus $M'_l=e_l$. We claim that $M_l\neq e_l$. Suppose the contrary. We compute M'_l.

$$e_l=M'_l=e_l(CM)=(e_lC)M=(M_k^{-1}+e_k+e_l)M$$

$$=(M_k^{-1})M+M_k+M_l=e_k+M_k+M_l.$$

Thus, if $M_l=e_l$, we have $e_l=e_k+M_k+e_l$, and hence $M_k=e_k$. But this contradicts the assumption that $k\in\mathcal{J}(M)$. Hence $M_l\neq e_l$ and so $l\in\mathcal{J}(M)$. Therefore,

$|\mathcal{J}(M')| = |\mathcal{J}| - 2$ and so by our induction hypothesis, $\sigma_{\{l\}}\pi_{M'} \in \Delta^{|\mathcal{J}|-2}$. Now

$$\pi_M = (\sigma_{\{k\}}\sigma_{\{k\}})\pi_C\pi_{M'} = (\sigma_{\{k\}}\pi_C)\sigma_{\{l\}}(\pi_{M'}).$$

Since both $\sigma_{\{k\}}\pi_C$ and $\sigma_{\{l\}}$ belong to Δ, it follows that $\pi_M \in \Delta^{2+(|\mathcal{J}|-2)} = \Delta^{|\mathcal{J}|}$.

Now suppose that $S \neq \emptyset$. We may assume that $k \in S$. Then $\sigma_S = \sigma_{\{k\}}\sigma_{S'}$, so $\sigma_S\pi_M = \sigma_{\{k\}}\sigma_{S'}\pi_C\pi_{M'}$, which is either $\sigma_{\{k\}}\pi_C\sigma_{S'}\pi_{M'}$ or $\sigma_{\{k\}}\sigma_{\{k,l\}}\pi_C\sigma_{S'}\pi_{M'} = \sigma_{\{l\}}\pi_C\sigma_{S'}\pi_{M'}$. Since both $\sigma_{\{k\}}\pi_C$ and $\sigma_{\{l\}}\pi_C$ belong to Δ, in either case we have $\sigma_S\pi_M \in \Delta^{1+|\mathcal{J}'|} = \Delta^{|\mathcal{J}|}$.

This completes the induction step, thereby proving the theorem. □

Example 4.9. Let

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Then $M^{-1} = M$. Note that $I + M = J$, the matrix all of whose entries are 1's. Hence $d(e_1, Me_1) = \mathit{weight}([1,1,1,1]) = 4$, and so $k_\Delta(\pi_M) = 4$. Now using the fact that $(M^{-1})_{11} = 0$ and $(M^{-1})_{12} = 1$, we have

$$C = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } M' = CM = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

$\pi_M = \pi_C\pi_{M'} = \pi_C(\sigma_{\{3\}}\sigma_{\{3\}})\pi_{M'} = (\pi_C\sigma_{\{3\}})(\sigma_{\{3\}}\pi_{M'})$. But $\sigma_{\{3\}}\pi_{M'} \in \Delta$ and $\pi_C\sigma_{\{3\}} = \sigma_{\{1,2,3\}}\pi_C = \sigma_{\{2,3\}}(\sigma_{\{1\}}\pi_C) \in \Delta^2 \cdot \Delta = \Delta^3$. Hence $\pi_M \in \Delta^4$.

Our next example shows that $k(\pi_M)$ can be less than $|\mathcal{J}(M)|$ and that π_M can be routed by LC permutations in fewer than $|\mathcal{J}(M)|$ steps.

Example 4.10. Let

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Gauss–Jordan elimination shows that M is invertible and that

$$M^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Now

$$I + M = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Since this has four nonzero rows, $|\mathcal{J}(M)| = 4$. On the other hand, row reduction shows that $(I + M)X = [1,1,1,1]^T$ has no solution, so that $[1,1,1,1]^T \notin \text{range}(I + M)$. It

follows from Lemma 3.5 that $k(\pi_M) \leq 3$. Since the third column of $I + M$ has weight 3, it follows that $k(\pi_M) = 3$.

Now by Lemma 4.7, part (2),

$$\pi_M = (\sigma_{\{3\}}\pi_{C_3})(\sigma_{\{2\}}\pi_{C_2})(\pi_{C_1}),$$

where $C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$, $C_2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, and $C_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

By Corollaries 3.3 and 3.8, each of these three factors belongs to Δ, and so we have a 3-step routing of π_M by LC permutations.

Remark. Removing the restriction to LC permutations allows us an alternate 3-step routing of π_M as follows. For compactness of notation, we shall denote each $x_1x_2x_3x_4$ by the integer $i \in \{0, 1, \ldots 15\}$ whose binary representation is $x_1x_2x_3x_4$. Then $\pi_M = (1, 12, 6, 11)(2, 9, 8, 4)(3, 5, 14, 15)(10, 13)$, and $\pi_M = \pi_3 \circ \pi_2 \circ \pi_1$, where

$$\pi_1 = (0, 2, 10, 8)(1, 5, 13, 9, 11, 3)(6, 14, 15, 7),$$

$$\pi_2 = (1, 5, 4, 6, 7, 3)(12, 14, 15, 13)(8, 9)(10, 11),$$

$$\pi_3 = (0, 4, 12, 14, 6, 2)(8, 10)(9, 13, 15, 11).$$

It is straightforward to check that for $1 \leq i \leq 3$, $k(\pi_i) = 1$ and so $\pi_i \in \Delta$.

PROPOSITION 4.11. *For any invertible matrix M, if $t(\pi_M)$ denotes the minimum number of steps in a congestion-free routing of π_M in which each factor is an LC permutation, then $2/3(|\mathcal{J}(M)|) \leq t(\pi_M) \leq |\mathcal{J}(M)|$.*

Proof. By Theorem 4.8, the minimum number of steps in a routing of π_M by LC permutations is less than or equal to $|\mathcal{J}(M)|$. Thus it suffices to show that if π_M is the product of j LC permutations, each of which is in Δ, then $|\mathcal{J}(M)| \leq 3j/2$. So suppose that $\pi_M = \varphi_j \circ \cdots \circ \varphi_2 \circ \varphi_1$, where each $\varphi_q \in \Delta$. Then $\varphi_q = \sigma_{A_q}\pi_{C_q}$, where either $A_q = \emptyset$ or $A_q = \{i_q\}$ and C_q differs from I in at most two rows, one of which is row i_q. Let $\mathcal{J}_q = \mathcal{J}(C_q)$. We claim first that there is at most one q such that $|\mathcal{J}_q| = 2$ and for all $p \neq q$, $\mathcal{J}_q \cap \mathcal{J}_p = \emptyset$. Suppose the contrary. Then for some q_1 and q_2, $\mathcal{J}_{q_1} = \{j_1, k_1\}$, $\mathcal{J}_2 = \{j_2, k_2\}$, and each of these sets is disjoint from the other and from $\mathcal{J}_p$ for all $p \neq q_1, q_2$. Since for $l = 1, 2$, $\varphi_{q_l} = \sigma_{A_{q_l}}\pi_{C_{q_l}} \in \Delta$, and $|\mathcal{J}_{q_l}| = 2$, we have $A_{q_l} = \{i_{q_l}\}$ and $i_{q_l} \in \mathcal{J}_{q_l}$. Say $i_{q_l} = j_l$. Since each $\mathcal{J}_{q_l}$ is disjoint from all other $\mathcal{J}_p$, it follows from Lemma 4.7 that $\pi_M = \sigma_{\{i_{q_1}, i_{q_2}\}}\sigma_B\pi_M = \sigma_{\{i_{q_1}, i_{q_2}\}\Delta B}\pi_M$, for some subset B disjoint from $\{i_{q_1}, i_{q_2}\}$, where Δ denotes the symmetric difference. Now the B in the representation of φ as $\sigma_B\pi_N$ is unique.

For $\varphi(0) = \sigma_B(0) = \sum_{i \in B} e_i$, and since the $\{e_i | 1 \leq i \leq n\}$ are a basis of $\mathcal{Z}_2^n$, this determines the set B. Hence $\{i_{q_1}, i_{q_2}\}\Delta B = \emptyset$, which is impossible since $\{i_{q_1}, i_{q_2}\}$ is disjoint from B. What this means is that the union of any pair of the subsets $\mathcal{J}_q$ has at most three elements. Since $\mathcal{J}(M) \subseteq \bigcup_{q=1}^{j} \mathcal{J}_q$, it follows that $|\mathcal{J}(M)| \leq 3j/2$. □

Remark. The lower bound in Proposition 4.11 is tight. For example, let $j = 3k$ and let the permutation θ of $\{1, 2, \ldots, j\}$ be the product of k disjoint 3-cycles, say $(1, 2, 3)(4, 5, 6) \cdots (3k - 2, 3k - 1, 3k)$.

Let P be the $j \times j$ permutation matrix whose ith row is $e_{\theta(i)}$. We claim that $\pi_P \in \Delta^{2k}$ so that $t(\pi_P) \leq 2k = 2j/3$. Proposition 4.11 gives the reverse inequality,

and so $t(\pi_P) = 2j/3$. Our claim will follow once we show that if θ' is any 3-cycle and if P' is the corresponding permutation matrix, then $\pi_{P'} \in \Delta^2$. To prove this claim, we may assume, with no loss in generality, that $\theta' = (1, 2, 3)$. Let P_{qr} denote the $j \times j$ permutation matrix obtained from I by interchanging rows q and r. Then $\pi_{P'} = (\sigma_{\{3\}}\pi_{P_{13}})(\sigma_{\{1\}}\pi_{P_{12}}) \in \Delta^2$.

Our final result is that every LC permutation has an n-step congestion-free routing.

COROLLARY 4.12. *Let M be any invertible $n \times n$ matrix and let T be any subset of $\{1, 2, \ldots, n\}$. Then $\sigma_T\pi_M \in \Delta^n$.*

Proof. Let $S = T \cap \mathcal{J}(M)$ and $T' = T \setminus S$. Then $\sigma_T\pi_M = \sigma_{T'}(\sigma_S\pi_M)$. By Theorem 4.8, $\sigma_S\pi_M \in \Delta^{|\mathcal{J}(M)|}$. Now $\sigma_{T'} \in \Delta^{|T'|}$ and $|T'| = |T| - |S|$. So $\sigma_T\pi_M \in \Delta^{|\mathcal{J}(M)|+|T|-|S|}$. But

$$|\mathcal{J}(M)| + |T| - |S| = |\mathcal{J}(M) \cup T| \leq n.$$

Hence $\sigma_T\pi_M \in \Delta^n$. ∎

REFERENCES

[1] R. BOPPANA AND C. RAGHAVENDRA, *Optimal self-routing of linear complement permutations in hypercubes*, in Proc. 5th Distributed Memory Computing Conf., IEEE Computer Society, Los Alamitos, CA, April 1990, pp. 800–808.

[2] F. T. LEIGHTON, *Introduction to Parallel Algorithms and Architectures: Arrays · Trees · Hypercubes*, Morgan-Kaufmann, San Mateo, CA, 1992.

[3] Z. LIU, AND J-H. YOU, *Conflict-free routing for BPC-permutations on synchronous hypercubes*, Parallel Comput., 19 (1993), pp. 323–342.

[4] D. NASSIMI AND S. SAHNI, *A self-routing Beneš network and parallel permutation algorithms*, IEEE Trans. Comput., C-30 (1981), pp. 332–340.

[5] D. NASSIMI AND S. SAHNI, *Optimal BPC permutations on a cube connected SIMD computer*, IEEE Trans. Comput., C-31 (1982), pp. 338–341.

[6] M. RAMRAS, *Routing permutations on a graph*, Networks, 23 (1993), pp. 391–398.

[7] M. RAMRAS, *Congestion-free optimal routings of hypercube automorphisms*, SIAM J. Discrete Math., 10 (1997), pp. 201–208.

[8] G. STRANG, *Linear Algebra and Its Applications*, Harcourt, Brace, and Jovanovich, Orlando, FL, 1988.

[9] K. ZEMOUDEH AND A. SENGUPTA, *Routing frequently used bijections on hypercube*, in Proc. 5th Distributed Memory Computing Conf., IEEE Computer Society, Los Alamitos, CA, April 1990, pp. 824–832.

SIAM J. DISCRETE MATH.
Vol. 11, No. 3, pp. 501–509, August 1998

011

FAULT-TOLERANT FIXED ROUTINGS IN SOME FAMILIES OF DIGRAPHS*

C. PADRÓ†, P. MORILLO†, AND X. MUÑOZ†

Abstract. The purpose of this paper is to find fault-tolerant fixed routings in some families of digraphs that have been widely considered into the design of interconnection networks. A routing ρ in a digraph G assigns to each pair of vertices a fixed path (called a route) between them. For a given set of faulty vertices and/or arcs, the vertices of the surviving route digraph are the nonfaulty vertices and there is an arc between two vertices if and only if there are no faults on the route between them. The diameter of the surviving route digraph measures the fault tolerance of the routing. In this work, sufficient conditions are found for a digraph to have a routing such that for any set of faults with a bounded number of elements the diameter of the surviving route digraph is at most 3. These results are applied to prove the existence of routings with this property in the generalized de Bruijn and Kautz digraphs, the bipartite digraphs $BD(d, n)$, and general iterated line digraphs.

Key words. fault-tolerant interconnection networks, fixed routings, surviving route digraph, generalized de Bruijn and Kautz digraphs, iterated line digraphs

AMS subject classifications. 68M10, 05C20, 05C38, 05C40

PII. S0895480195280940

1. Introduction.

Interconnection networks are usually modeled by graphs, directed or not, in which the vertices represent the switching elements or processors. Communication links are represented by edges if they are bidirectional and by arcs if they are unidirectional. Communication between nodes can be done through paths that have been fixed in advance, especially if the nodes have no information about the topology of the network.

The reader is referred to Chartrand and Lesniak [3] for the graph theoretical concepts which are not defined here. Directed graphs are usually called *digraphs* for short. Here we are concerned with digraphs only. All digraphs will be supposed to be strongly connected. In this paper, a *path of length* t between vertices x and y of a digraph G will be a sequence $x_0 = x, x_1, \ldots, x_{t-1}, x_t = y$, where (x_{i-1}, x_i) is an arc of G. The vertices and arcs in a path are not necessarily different. For any set A of vertices of a digraph G, let $\Gamma_l^+(A)$ be defined recursively by $\Gamma_l^+(A) = \Gamma^+(\Gamma_{l-1}^+(A))$ beginning with

$$\Gamma_1^+(A) = \Gamma^+(A) = \bigcup_{v \in A} \Gamma^+(v),$$

where $\Gamma^+(v)$ is the set of vertices adjacent from v. When $A = \{v\}$ we just write $\Gamma_l^+(v)$. If $\Gamma^-(v)$ denotes the set of all vertices adjacent to v, we can define analogously the sets $\Gamma_l^-(A)$.

A *routing* ρ in a graph or digraph G assigns to every pair of different vertices a path $\rho(x, y)$ from x to y. The paths $\rho(x, y)$ are called *routes*. We assume that all communications between vertices are done through the routes of a fixed routing.

*Received by the editors February 1, 1995; accepted for publication (in revised form) June 2, 1997; published electronically July 7, 1998. This work was supported by CICYT under project TIC 94-0760.

http://www.siam.org/journals/sidma/11-3/28094.html

†Departament de Matemàtica Aplicada i Telemàtica, Universitat Politècnica de Catalunya, Campus Nord, Edifici C3, C/Gran Capita, s/n, E-08034 Barcelona, Spain (matcpl@mat.upc.es, paz@mat.upc.es, xml@mat.upc.es).

Two parameters have been proposed to measure the efficiency and fault tolerance of a fixed routing in a graph or a digraph: the forwarding index, introduced by Chung et al. [4] and the diameter of the surviving route digraph, proposed by Dolev et al. [5].

The *vertex-forwarding index* of a routing ρ in a graph or digraph G, $\xi(G,\rho)$, is the maximum number of routes passing through a vertex. The *edge-* or *arc-forwarding index*, $\pi(G,\rho)$, is defined analogously. In order to construct efficient routings, we have to minimize the forwarding index. That is, the routes should not load a node or a link too much: too many routes should not go through it. If the routing is well distributed, that is, all vertices and edges or arcs have a similar load, queues will be shortened and the failure of a node or a link will not destroy too many routes.

When some nodes and/or links of the network fail, the routes containing faulty elements cannot be used. However, perhaps the communication can still be possible by using a sequence of routes not containing faulty elements. For a given set F of faulty vertices and/or arcs, the vertices of the *surviving route digraph* $R(G,\rho)/F$ are the non-faulty vertices and there is an arc between two vertices if and only if there are no faults on the route between them. Fault-tolerant routings are such that the diameter of the surviving route digraph, $D(R(G,\rho)/F)$, is small for any set of faults of bounded size.

Because of the good relation between their order, degree, and diameter, some families of digraphs have been specially considered into the design of interconnection networks. The largest known (d,D)-digraphs (digraphs with maximum out-degree d and diameter D) are the *de Bruijn* [2] and *Kautz* [14] *digraphs*, denoted, respectively, by $B(d,D)$ and $K(d,D)$. The diameter of *Reddy–Pradhan–Kuhl or generalized de Bruijn digraphs*, $RPK(d,n)$ [19, 18, 12], and *Imase–Itoh or generalized Kautz digraphs*, $II(d,n)$ [13], are minimum or quasi-minimum for their degree and order. Fiol and Yebra [9] introduced a family of bipartite digraphs, the *bipartite digraphs* $BD(d,n)$, with minimum or quasi-minimum diameter. The digraphs $BD(d,d^{D-1}+d^{D-3})$ are large (d,D)-bipartite digraphs (optimal if $D \leq 5$). The connectivity of all these digraphs is optimal in most cases [1, 9].

Kautz and de Bruijn digraphs and the bipartite digraphs $BD(d,d^{D-1}+d^{D-3})$ are iterated line digraphs [8, 9]. We recall here the definition and some properties of line digraphs. See, for example, [10] for proofs and more information.

In the *line digraph* LG of a digraph G each vertex represents an arc of G, that is, $V(LG) = \{uv \,|\, (u,v) \in A(G)\}$. A vertex uv is adjacent to a vertex wz if and only if $v = w$, that is, whenever the arc (u,v) of G is adjacent to the arc (w,z). The maximum and minimum out- and in-degrees of LG are equal to those of G. Therefore, if G is d-regular with order n, then LG is d-regular and has order dn. If G is a strongly connected digraph different from a directed cycle, then the diameter of LG is the diameter of G plus one.

The iteration of the line digraph operation is a good method to obtain large digraphs with fixed degree and diameter. If G is d-regular, and has diameter D and order n, then L^kG is d-regular, and has diameter $D+k$ and order d^kn; that is, the order increases in an asymptotically optimal way in relation to the diameter. Routings in the iterated line digraph L^kG can be easily derived from those in G. Besides, iterated line digraphs have maximum connectivity if the number of iterations is large enough [7]. The diameter-vulnerability, that is, the maximum diameter after deleting a fixed number of vertices or arcs, of the iterated line digraphs L^kG is independent from the number of iterations [16].

Homobono and Peyrat [11] and, in a different way, Escudero et al. [6] prove that

for any routing of shortest paths in the Kautz and de Bruijn digraphs and for any set of faults of cardinality $\kappa(G)-1$ ($\kappa(G)$ is the minimum number of vertices whose deletion disconnects the digraph), the diameter of the surviving route digraph is 2. Manabe, Imase, and Soneoka [15] find routings in the generalized de Bruijn digraphs, $RPK(d,n)$, with small forwarding index and diameter of the surviving route digraph at most 3. Routings on the bipartite digraphs $BD(d, d^{D-1}+d^{D-3})$ with almost optimal forwarding index and diameter of the surviving route digraph equal to 2 are given in [17]. A bound for the diameter of the surviving route digraph of general iterated line digraphs is given in [6]. It is proven in [6] that for any loopless digraph G, there are fixed routings ρ in L^kG such that, if k is large enough, the diameter of the surviving route digraph is at most $2(D(G)-R(G)+1)$, where $D(G)$ is the diameter of G and $R(G)$ is a parameter that, in most cases, is lesser than the diameter $D(G)$. Therefore, $2(D(G)-R(G)+1)>3$ except for some particular graphs.

The purpose of this paper is to find fault-tolerant routings in the generalized de Bruijn and Kautz digraphs, the bipartite digraphs $BD(d,n)$, and in general iterated line digraphs.

We present in section 2 sufficient conditions for a digraph to have a routing such that, for any set of faults of bounded size, the diameter of the surviving route digraph is at most 3. Using this condition, we present in sections 3 and 4 routings with this property in the generalized de Bruijn and Kautz digraphs, the bipartite digraphs $BD(d,n)$, and general iterated line digraphs, loopless or not.

2. Sufficient conditions. Sufficient conditions for a digraph to have a routing with diameter of the surviving route digraph at most three are given in this section. These conditions are given in terms of the maximum and minimum degrees and two parameters, h and r, which will be defined later.

Let $G=(V,A)$ be a digraph, ρ a routing in G, and $F\subset V\cup A$, $|F|<\kappa(G)$, a set of faults. Let F_ρ be the set of all routes $\rho(x,y)$ containing items in F. If $x\notin F$ is a vertex in G, $d_R^-(x)$ and $d_R^+(x)$ will stand for the in-degree and the out-degree of vertex x in the surviving route digraph $R(G,\rho)/F$. Let us note that $d_R^+(x)$ is the number of vertices z such that route $\rho(x,z)$ avoids F. Let $\mathrm{d}_R(x,y)$ be the distance between nonfaulty vertices x and y in the surviving route digraph.

PROPOSITION 2.1. *Let $G=(V,A)$ be a digraph and ρ a routing in G. Let $F\subset V\cup A$, $|F|<\kappa(G)$, be a set of faults. Let x and y be two different nonfaulty vertices such that $(d_R^+(x)+1)(d_R^-(y)+1)>|F_\rho|$. Then $\mathrm{d}_R(x,y)\le 3$.*

Proof. Let $R^+(x)$ be the set containing vertex x and all adjacent vertices from x in the surviving route digraph. This set has exactly $d_R^+(x)+1$ elements. Analogously, $d_R^-(y)+1$ denote the number of vertices in the set $R^-(y)$, which contains the vertex y and all vertices adjacent to y in the surviving route digraph. Therefore, if $(d_R^+(x)+1)(d_R^-(y)+1)>|F_\rho|$, then $R^+(x)\cap R^-(x)\neq\emptyset$ or there must exist vertices $z_1\in R^+(x)$ and $z_2\in R^-(y)$ such that $\rho(z_1,z_2)\notin F_\rho$. Hence, $\mathrm{d}_R(x,y)\le 3$. □

We will present bounds on $(d_R^+(x)+1)(d_R^-(y)+1)$ and on $|F_\rho|$ for a special kind of routings. Comparing these bounds, sufficient conditions for a digraph to have a routing with diameter of the surviving route digraph at most three will be found. First, we are going to define the parameters h and r and the routings we are going to consider.

Let G be a digraph with diameter D. Let us define $h=h(G)$, $1\le h\le D$, as the maximum integer such that if x and y are two (not necessarily different) vertices in G, there cannot exist two different paths from x to y with the same length $t\le h$. Notice that if G is d-regular, $h(G)$ is the maximum integer such that for all $t\le h$ and

for all vertex x, $|\,\Gamma_t^+(x)\,| = d^t$.

Let $G = (V, A)$ be a digraph with maximum degree Δ and $h = h(G)$. Let us define $r = r(G)$ as the minimum integer such that for each vertex x in G,

$$V = \{\,x\,\} \cup \Gamma_h^+(x) \cup \Gamma_{h+1}^+(x) \cup \cdots \cup \Gamma_{h+r}^+(x).$$

Let ρ be a routing in G. We will say that ρ is an *h-routing* if the length t of any route $\rho(x, y)$ is such that $h \le t \le h + r$ and the route $\rho(x, y)$ is the only path of length h from x to y if $y \in \Gamma_h^+(x)$.

PROPOSITION 2.2. *Let $G = (V, A)$ be a digraph and ρ a routing in G with vertex-forwarding index $\xi = \xi(G, \rho)$ and arc-forwarding index $\pi = \pi(G, \rho)$. Let $F \subset V \cup A$, $|\,F\,| < \kappa(G)$, be a set of faults. Then*

$$|\,F_\rho\,| \le |\,F \cap V\,|\xi + |\,F \cap A\,|\pi.$$

PROPOSITION 2.3. *Let G be a digraph with maximum degree Δ, $h = h(G)$, and $r = r(G)$. Let ρ be an h-routing in G. Then $\xi(G, \rho) \le \Theta(\Delta, h, r)$ and $\pi(G, \rho) \le \Pi(\Delta, h, r)$, where*

$$\Theta(\Delta, h, r) = (h-1)\Delta^h + h\Delta^{h+1} + \cdots + (h+r-1)\Delta^{h+r}$$

and

$$\Pi(\Delta, h, r) = (\Theta(\Delta, h, r) + P(\Delta, h, r))/\Delta,$$

with $P(\Delta, h, r) = \Delta^h + \Delta^{h+1} + \cdots + \Delta^{h+r}$.

Proof. The number of paths of length t passing through each vertex is at most $(t-1)\Delta^t$. The number of paths of length t passing through an arc is at most $t\Delta^{t-1}$. □

PROPOSITION 2.4. *For any integers $\delta \ge 3$ and $h \ge 4$, let us consider*

1. $m_0(\delta, h) = \frac{1}{(\delta-1)^2}((\delta-2)\delta^{2h} - (\delta-3)\delta^s)$,
2. $m_1(\delta, h) = \frac{1}{(\delta-1)^2}((\delta-3)\delta^{2h} - (\delta-4)\delta^s)$ *if $\delta \ge 4$, and*
3. $m_1(3, h) = \frac{3^{2h}}{4} + \frac{3^h}{2}$,

where $s = 3h/2$ if h is even and $s = (3h+1)/2$ if h is odd. Let $G = (V, A)$ be a digraph with minimum degree $\delta \ge 3$ and $h = h(G) \ge 4$. Let ρ be an h-routing in G and let $F \subset V \cup A$, $|\,F\,| \le \delta - 2$, be a set of faults. Let us consider two different vertices $x, y \notin F$. Then $(d_R^+(x)+1)(d_R^-(y)+1) \ge m_0(\delta, h)$ if F has no arcs. If the only arc in F is (x, y), then $(d_R^+(x)+1)(d_R^-(y)+1) \ge m_1(\delta, h)$.

Proof. We are going to consider only h even. If h is odd, the proof is similar.

Let us suppose F does not contain any arc. For $i = 1, 2, \ldots, h$ let us consider $\mu_i = |\,\Gamma_i^+(x) \cap F\,|$ and $\nu_i = |\,\Gamma_i^-(y) \cap F\,|$. Obviously, $\mu_i, \nu_i \le \delta - 2$. Moreover, if $i + j \le h$, $\mu_i + \nu_j \le |\,F\,| + 1 \le \delta - 1$. Certainly, since there cannot exist two different paths of length $i + j$ from x to y, the intersection $\Gamma_i^+(x) \cap \Gamma_j^-(y)$ contains at most one vertex. Therefore, only one item in F can be counted twice in $\mu_i + \nu_j$.

In order to find a lower bound for the number of routes $\rho(x, z)$ of length h avoiding F, we can suppose that all vertices have out-degree equal to the minimum degree δ. That is, we are going to ignore $d^+(v) - \delta$ arcs for each vertex v. Analogously, we are going to suppose that all vertices have in-degree equal to δ when calculating a lower bound for the number of routes $\rho(z, y)$ of length h avoiding F.

Therefore, the number of routes $\rho(x, z)$ of length h containing a faulty vertex in position i is at most $\mu_i\delta^{h-i}$ and the number of routes $\rho(x, z)$ of length h avoiding F

is greater than or equal to $\delta^h - \sum_{i=1}^{h} \mu_i \delta^{h-i} - 1$. Notice that it is possible $x \in \Gamma_h^+(x)$. Therefore,

$$d_R^+(x) + 1 \geq \delta^h - \sum_{i=1}^{h} \mu_i \delta^{h-i}.$$

Analogously,

$$d_R^-(y) + 1 \geq \delta^h - \sum_{i=1}^{h} \nu_i \delta^{h-i}.$$

Let us define $X = \sum_{i=1}^{h} \mu_i \delta^{h-i}$ and $Y = \sum_{i=1}^{h} \nu_i \delta^{h-i}$. Since $\mu_i, \nu_i \leq \delta - 2$, we have $0 \leq X, Y \leq M = (\delta - 2)(\delta^h - 1)/(\delta - 1)$. Furthermore,

$$X + Y = \sum_{i=1}^{h} (\mu_i + \nu_i)\delta^{h-i} \leq (\delta - 1)(\delta^{h-1} + \cdots + \delta^{h/2}) + 2(\delta - 2)(\delta^{(h-2)/2} + \cdots + 1).$$

Hence, $X + Y \leq S$ with

$$S = \delta^h - \delta^{h/2} + 2(\delta - 2)\frac{\delta^{h/2} - 1}{\delta - 1}.$$

With these restrictions, the minimum value of $(\delta^h - X)(\delta^h - Y)$ is attained when $X = M$ and $Y = S - M$ or vice versa. Therefore,

$$(d_R^+(x) + 1)(d_R^-(y) + 1) \geq (\delta^h - M)(\delta^h - S + M),$$

and the proof of this case is finished with a straightforward calculation.

If $F \cap A = \{(x, y)\}$, we consider $F_1 = (F \cap V) \cup \{y\}$ and $F_2 = (F \cap V) \cup \{x\}$. Notice that all routes $\rho(x, z)$ containing the arc (x, y) pass through the vertex y. Therefore, if $\mu_i = |\, \Gamma_i^+(x) \cap F_1 \,|$, we have

$$d_R^+(x) + 1 \geq \delta^h - \sum_{i=1}^{h} \mu_i \delta^{h-i}.$$

Analogously,

$$d_R^-(y) + 1 \geq \delta^h - \sum_{i=1}^{h} \nu_i \delta^{h-i},$$

where $\nu_i = |\, \Gamma_i^-(y) \cap F_2 \,|$. In this case, $\mu_i, \nu_i \leq \delta - 2$ and $\mu_i + \nu_j \leq |\, F \,| + 2 \leq \delta$ if $i + j \leq h$. But if $\delta = 3$, then $F \cap V = \emptyset$ and $\mu_i + \nu_i \leq 2$. From this point the proof follows with the same arguments used in the case before. □

PROPOSITION 2.5. *For any pair of integers $\delta \geq 2$ and $h \geq 4$, we consider*

1. $n_0(\delta, h) = (\delta^3 - 3\delta^2 + 4\delta - 3)\delta^{2h-4} - (\delta - 2)\delta^s$,
2. $n_1(\delta, h) = (\delta^3 - 3\delta^2 + 4\delta - 4)\delta^{2h-4} - (\delta - 3)\delta^s$ *if* $\delta \geq 3$ *and*
3. $n_1(2, h) = 2^{2h-4} + 2^{h-1} + 1$,

where $s = (3h - 3)/2$ if h is odd and $s = (3h - 4)/2$ if h is even. Let $G = (V, A)$ be a loopless digraph with minimum degree $\delta \geq 2$ and $h = h(G) \geq 4$ such that, for any vertex x, $\Gamma^+(x) \cap \Gamma_2^+(x) = \emptyset$. Let ρ be an h-routing in G. Let $F \subset V \cup A$,

$|F| = \delta - 1$, *be a set of faulty items and* $x, y \notin F$ *a pair of nonfaulty vertices. Then* $(d_R^+(x)+1)(d_R^-(y)+1) \geq n_0(\delta,h)$ *if* F *has no arcs. If* $F \cap A = \{(x,y)\}$, *then* $(d_R^+(x)+1)(d_R^-(y)+1) \geq n_1(\delta,h)$.

Proof. The proof of this proposition follows the same way as in Proposition 2.4. The only difference is that we have to take into account that $|F| = \delta - 1$ and that $\mu_1 + \mu_2 \leq \delta - 1$ and $\nu_1 + \nu_2 \leq \delta - 1$ because there cannot exist any path of length 2 between adjacent vertices. □

If we want to find upper bounds for the distance in the surviving route digraph from a vertex x to a vertex y, we can suppose that F has no arcs or the only arc in F is (x,y). Actually, let $(u,v) \neq (x,y)$ be an arc in F. If $u \neq x$, let us consider $F' = (F - \{(u,v)\}) \cup \{u\}$. If $u = x$, we take $F' = (F - \{(u,v)\}) \cup \{v\}$. It is easy to see that $\mathrm{d}_R(x,y) \leq \mathrm{d}_{R'}(x,y)$ if $R' = R(G,\rho)/F'$

THEOREM 2.6. *Let* G *be a digraph with minimum degree* $\delta \geq 3$, *maximum degree* Δ, $h = h(G) \geq 4$ *and* $r = r(G)$ *such that* $m_0(\delta,h) > (\delta-2)\Theta(\Delta,h,r)$ *and* $m_1(\delta,h) > (\delta-3)\Theta(\Delta,h,r) + \Pi(\Delta,h,r)$. *Then, for any* h*-routing* ρ *and for any set of faults* F, $|F| \leq \delta - 2$, *the diameter of the surviving route digraph* $R(G,\rho)/F$ *is at most* 3.

Proof. We have to prove that for any pair of different vertices x, y of G, $d_R(x,y) \leq 3$.

If F does not contain the arc (x,y), we can suppose that F contains no arcs. Therefore, from Propositions 2.2, 2.3, and 2.4,

$$(d_R^+(x)+1)(d_R^-(y)+1) \geq m_0(\delta,h) > (\delta-2)\Theta(\Delta,h,r) \geq |F_\rho|.$$

Hence, from Proposition 2.1, $d_R(x,y) \leq 3$.

If (x,y) is a faulty arc, we can suppose that $F \cap A = \{(x,y)\}$. In this case,

$$(d_R^+(x)+1)(d_R^-(y)+1) \geq m_1(\delta,h) > (\delta-3)\Theta(\Delta,h,r) + \Pi(\Delta,h,r) \geq |F_\rho|. \qquad \square$$

THEOREM 2.7. *Let* G *be a loopless digraph with minimum degree* $\delta \geq 2$, *maximum degree* Δ, $h = h(G) \geq 4$, *and* $r = r(G)$ *such that* $\Gamma^+(x) \cap \Gamma_2^+(x) = \emptyset$ *for any vertex* x, $n_0(\delta,h) > (\delta-1)\Theta(\Delta,h,r)$ *and* $n_1(\delta,h) > (\delta-2)\Theta(\Delta,h,r) + \Pi(\Delta,h,r)$. *Then, for any* h*-routing* ρ *and for any set of faults* F, $|F| \leq \delta - 1$, *the diameter of the surviving route digraph* $R(G,\rho)/F$ *is at most* 3.

Proof. The proof of this theorem is similar to that of Theorem 2.6. □

3. Routings in generalized Kautz and de Bruijn digraphs and bipartite digraphs $BD(d,n)$**.** For any integers $n \geq d \geq 2$, the *Reddy–Pradhan–Kuhl or generalized de Bruijn digraph with degree* d *and order* n, $RPK(d,n)$ [19, 18, 12], has a set of vertices $\mathbf{Z}_n$. The arcs of $RPK(d,n)$ are in the form $(x, dx+t)$, where $0 \leq t \leq d-1$. This digraph is d-regular and has diameter $D = \lceil \log_d n \rceil$. The digraph $RPK(d,d^D)$ is isomorphic to the de Bruijn digraph $B(d,D)$.

The vertices of the *Imase–Itoh or generalized Kautz digraph with degree* d *and order* n, $II(d,n)$ [13], are the elements of $\mathbf{Z}_n$. A vertex x of $II(d,n)$ is adjacent to the vertices $-dx - t$ for any $t = 1, \ldots, d$. This digraph is d-regular and its diameter is such that

$$\lfloor \log_d n \rfloor \leq D \leq \lceil \log_d n \rceil.$$

The Kautz digraph $K(d,D)$ coincides with the digraph $II(d, d^D + d^{D-1})$.

The *bipartite digraphs* $BD(d,n)$ [9] are defined by taking as set of vertices $V = \{0,1\} \times \mathbf{Z}_n$ and adjacencies

$$\Gamma^+(\alpha,x) = \{(\overline{\alpha}, (-1)^\alpha d(x+\alpha) + t) \,|\, t = 0, 1, \ldots, d-1\}.$$

TABLE 3.1
Values of d and h for Theorem 3.2.

$d = 2$	and	$h \geq 9$
$d = 3$	and	$h \geq 6$
$4 \leq d \leq 6$	and	$h \geq 5$
$d \geq 7$	and	$h \geq 4$

TABLE 3.2
Values of d and h for Theorem 3.3.

$d = 2$	and	$h \geq 10$
$d = 3$	and	$h \geq 7$
$4 \leq d \leq 8$	and	$h \geq 6$
$d \geq 9$	and	$h \geq 5$

The digraph $BD(d, n)$ is bipartite and d-regular and has diameter D such that

$$\lfloor \log_d n \rfloor + 1 \leq D \leq \lceil \log_d n \rceil + 1.$$

The next proposition is a direct consequence of the properties of these digraphs, which are given in [12, 13, 9].

PROPOSITION 3.1. *If G is $RPK(d, n)$, $II(d, n)$ or $BD(d, n)$, then $h(G) = \lfloor \log_d n \rfloor$. If G is $RPK(d, n)$ or $II(d, n)$, $r(G) \leq 1$ and $r(G) \leq 2$ if $G = BD(d, n)$.*

THEOREM 3.2. *Let $G = (V, A)$ be one of $RPK(d, n)$ or $II(d, n)$, $d \geq 3$, such that $h = h(G) \geq 4$. Let ρ be an h-routing in G. Then, for every set of faults $F \subset V \cup A$, $|F| \leq d - 2$, the diameter of the surviving route digraph $R(G, \rho)/F$ is at most 3.*

Proof. Applying Theorem 2.6 for $\delta = \Delta = d$ and $r = 1$, it is enough to check that $m_0(d, h) > (d - 2)\Theta(d, h, 1)$ and $m_1(d, h) > (d - 3)\Theta(d, h, 1) + \Pi(d, h, 1)$ if $h \geq 4$. □

THEOREM 3.3. *Let $d \geq 2$ and let n be a multiple of $d(d + 1)$. Let G be the Imase–Itoh digraph $II(d, n)$, and let ρ be an h-routing in G. Then, for the values of h and d given in Table 3.1 and for all set of failures F, $|F| \leq d - 1$, the diameter of the surviving route digraph $R(G, \rho)/F$ is at most 3.*

Proof. Since n is a multiple of $d(d + 1)$, for any vertex x of the Imase–Itoh digraph $II(d, n)$, $\Gamma^+(x) \cap \Gamma_2^+(x) = \emptyset$. Therefore, we can apply Theorem 2.7 for $\delta = \Delta = d$ and $r = 1$. We have only to check that $n_0(d, h) > (d - 1)\Theta(d, h, 1)$ and $n_1(d, h) > (d - 2)\Theta(d, h, 1) + \Pi(d, h, 1)$ if d and h are in Table 3.1. □

THEOREM 3.4. *Let G be the bipartite digraph $BD(d, n)$ and let ρ be an h-routing in G. Then, for the values of h and d given in Table 3.2 and for all set of failures F, $|F| \leq d - 1$, the diameter of the surviving route digraph $R(G, \rho)/F$ is at most 3.*

Proof. Since G is bipartite, it is obvious that for any vertex x, $\Gamma^+(x) \cap \Gamma_2^+(x) = \emptyset$. Therefore, Theorem 2.7 can be applied for $\delta = \Delta = d$ and $r = 2$. The proof is finished by checking that $n_0(d, h) > (d-1)\Theta(d, h, 2)$ and $n_1(d, h) > (d-2)\Theta(d, h, 2)+\Pi(d, h, 2)$ for the values of d and h in Table 3.2. □

4. Routings in iterated line digraphs. The next proposition is proved using the properties of iterated line digraphs given in [10].

PROPOSITION 4.1. *Let G be a digraph with minimum degree $\delta \geq 2$ and maximum degree Δ. Then, for any $k \geq 1$, the iterated line digraph L^kG has minimum degree δ and maximum degree Δ, $h(L^kG) = h(G) + k$ and $r(L^kG) = r(G)$. Besides, if G is loopless, for any vertex x of L^kG, $\Gamma^+(x) \cap \Gamma_2^+(x) = \emptyset$.*

PROPOSITION 4.2. *For any Δ, h, and r and for any $k \geq 1$,*

1. $\Theta(\Delta, h+k, r) = \Delta^k \Theta(\Delta, h, r) + k\Delta^k P(\Delta, h, r)$.
2. $\Pi(\Delta, h+k, r) = \Delta^{k-1}\Theta(\Delta, h, r) + (k+1)\Delta^{k-1} P(\Delta, h, r)$.

Proof. We can easily prove this proposition from the definitions of Θ, Π, and P. □

THEOREM 4.3. *Let $G = (V, A)$ be a digraph with minimum degree $\delta \geq 3$ and maximum degree Δ such that $\Delta < \delta^2$. Then, if k is large enough, for any h-routing in the iterated line digraph L^kG and for any set of faults F, $|F| \leq \delta - 2$, the diameter of the surviving route digraph $R(L^kG, \rho)/F$ is at most 3. Besides, if G is a loopless digraph and k is large enough, for any h-routing in L^kG and for any set of faults F, $|F| = \delta - 1$, $D(R(L^kG, \rho)/F) \leq 3$.*

Proof. Let us consider $\Theta = \Theta(\Delta, h, r)$ and $P = P(\Delta, h, r)$. It is not difficult to prove that

$$\lim_{k\to\infty} \frac{(\delta-2)\Theta(\Delta, h+k, r)}{m_0(\delta, h+k)} = \lim_{k\to\infty} \frac{(\delta-2)(\Delta^k\Theta + k\Delta^k P)}{m_0(\delta, h+k)} = 0.$$

It can be also proved that

$$\lim_{k\to\infty} \frac{(\delta-3)\Theta(\Delta, h+k, r) + \Pi(\Delta, h+k, r)}{m_1(\delta, h+k)} = 0.$$

Therefore, from Theorem 2.6, the diameter of the surviving route digraph $R(L^kG, \rho)/F$ is at most 3 if k is large enough.

The case of loopless digraphs is proved analogously. □

REFERENCES

[1] J. C. BERMOND, N. HOMOBONO, AND C. PEYRAT, *Large fault-tolerant interconnection networks*, Graphs Combin., 5 (1989), pp. 107–123.

[2] N. G. DE BRUIJN, *A combinatorial problem*, Konink. Nederl. Akad. Wetensch. Verh. Afd. Natuurk. Eerste Reeks, A49 (1946), pp. 758–764.

[3] G. CHARTRAND AND L. LESNIAK, *Graphs & Digraphs*, Wadsworth & Brooks, Monterey, CA, 1986.

[4] F. R. K. CHUNG, E. G. COFFMAN JR., M. I. REIMAN, AND B. SIMON, *On the capacity and forwarding index of communication networks*, IEEE Trans. Inform. Theory, 33 (1987), pp. 224–232.

[5] D. DOLEV, J. HALPERN, B. SIMONS, AND H. STRONG, *A new look at fault-tolerant network routing*, Inform. and Comput., 72 (1987), pp. 180–196.

[6] M. ESCUDERO, J. FÀBREGA, M. A. FIOL, AND N. HOMOBONO, *On surviving route graphs of iterated line digraphs*, Graph Theory, Combinatorics and Applications, 1 (1988), pp. 451–466.

[7] J. FÀBREGA AND M. A. FIOL, *Maximally connected digraphs*, J. Graph Theory, 13 (1989), pp. 657–668.

[8] M. A. FIOL, I. ALEGRE, AND J. L. A. YEBRA, *Line digraph iterations and (d, k) problem for directed graphs*, in Proc. 10th Int. Symp. Comput. Arch., 1983, pp. 174–177.

[9] M. A. FIOL AND J. L. A. YEBRA, *Dense bipartite digraphs*, J. Graph Theory, 14 (1990), pp. 687–700.

[10] M. A. FIOL, J. L. A. YEBRA, AND I. ALEGRE, *Line-digraph iterations and the (d, k) problem*, IEEE Trans. Comput., C-33 (1984), pp. 400–403.

[11] N. HOMOBONO AND C. PEYRAT, *Fault tolerant routings in Kautz and de Bruijn networks*, in Combinatorial Conference, Montreal, 1987.

[12] M. Imase and M. Itoh, *Design to minimize diameter on building-block network*, IEEE Trans. Comput., C-30 (1981), pp. 439–442.
[13] M. Imase and M. Itoh, *A design for directed graphs with minimum diameter*, IEEE Trans. Comput., C-32 (1983), pp. 782–784.
[14] W. H. Kautz, *Bounds on directed* (d, k) *graphs*, Theory of cellular logic networks and machines, AFCRL-68-0668 Final Report, 1968, pp. 20–28.
[15] Y. Manabe, M. Imase, and T. Soneoka, *Reliable and efficient fixed routings on digraphs*, Transactions of the Institute of Electronics, Information and Communication Engineers E, E71 (1988), pp. 1212–1220.
[16] C. Padró and P. Morillo, *Diameter-vulnerability of iterated line digraphs*, Discrete Math., 149 (1996), pp. 189–204.
[17] C. Padró, P. Morillo, and E. Llobet, *Efficient and fault-tolerant fixed routings on bipartite digraphs*, J. Combin. Inform. System Sci., to appear.
[18] S. M. Reddy, D. K. Pradhan, and J. G. Kuhl, *Directed Graphs with Minimal Diameter and Maximum Node Connectivity*, Tech. report, School of Engineering, Oakland University, Rochester, MI, 1980.
[19] M. L. Schlumberger, *De Bruijn Communication Networks*, Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA, 1974.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 511–523, November 1998

A GEOMETRIC APPROACH TO BETWEENNESS*

BENNY CHOR† AND MADHU SUDAN‡

Abstract. An input to the *betweenness* problem contains m constraints over n real variables (points). Each constraint consists of three points, where one of the points is specified to lie inside the interval defined by the other two. The order of the other two points (i.e., which one is the largest and which one is the smallest) is not specified. This problem comes up in questions related to physical mapping in molecular biology. In 1979, Opatrny showed that the problem of deciding whether the n points can be totally ordered while satisfying the m betweenness constraints is NP-complete [*SIAM J. Comput.*, 8 (1979), pp. 111–114]. Furthermore, the problem is MAX SNP complete, and for every $\alpha > 47/48$ finding a total order that satisfies at least α of the m constraints is NP-hard (even if all the constraints are satisfiable). It is easy to find an ordering of the points that satisfies $1/3$ of the m constraints (e.g., by choosing the ordering at random).

This paper presents a polynomial time algorithm that either determines that there is no feasible solution or finds a total order that satisfies at least $1/2$ of the m constraints. The algorithm translates the problem into a set of quadratic inequalities and solves a semidefinite relaxation of them in $\mathcal{R}^n$. The n solution points are then projected on a random line through the origin. The claimed performance guarantee is shown using simple geometric properties of the semidefinite programming (SDP) solution.

Key words. approximation algorithm, semidefinite programming, NP-completeness, computational biology

AMS subject classifications. 68Q20, 68Q25

PII. S0895480195296221

1. Introduction.

An input to the *betweenness* problem consists of a finite set of n elements (or *points*) $S = \{x_1, \ldots, x_n\}$ and a finite set of m constraints. Each constraint consists of a triplet $(x_i, x_j, x_k) \in S \times S \times S$. A candidate solution to the betweenness problem is a total order $<$ on its points. A total order $x_{i_1} < x_{i_2} < \cdots < x_{i_n}$ satisfies the constraint (x_i, x_j, x_k) if either $x_i < x_j < x_k$ or $x_k < x_j < x_i$. That is, each constraint forces the second variable x_j to be between the other two variables x_i and x_k but does not specify the relative order of x_i and x_k. The decision version of the betweenness problem is to decide if all constraints can be simultaneously satisfied by a total order of the variables.

In 1979, Opatrny [14] showed that the decision version of the betweenness problem is NP-complete. This problem arises naturally when analyzing certain mapping problems in molecular biology. For example, it arises when trying to order markers on a chromosome, given the results of a radiation hybrid experiment [6, 3]. A computational task of practical significance in this context is to find a total ordering of the markers (the x_i in our terminology) that maximizes the number of satisfied constraints. Indeed, betweenness is central in the recent software package RHMAPPER

*Received by the editors December 18, 1995; accepted for publication (in revised form) February 25, 1998; published electronically September 1, 1998. A preliminary version of this paper appeared in the *Proceedings of the Third Annual European Symposium on Algorithms* (*ESA* '95), Lecture Notes in Comput. Sci. 979, Paul Spirakis, ed., Springer-Verlag, Berlin, New York, Heidelberg, 1995, pp. 227–237.

http://www.siam.org/journals/sidma/11-4/29622.html

†Department of Computer Science, Technion, Haifa 32000, Israel (benny@cs.technion.ac.il). The research of this author was partially supported by Technion V.P.R. funds.

‡Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139 (madhu@lcs.mit.edu). Part of this work was done while this author was at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY.

[15, 16]. At the heart of this package is a method for producing the order of *framework markers* based on betweenness constraints (obtained from a statistical analysis of the biological data). Slonim et al. [16]. successfully employ two greedy heuristics for solving the betweenness problem.

Opatrny gave two reductions in his proof of NP-completeness. One of these reductions is from 3SAT. Following his construction, we show in section 2 an approximation preserving reduction from MAX 3SAT. This implies that there exists an $\varepsilon > 0$, such that finding a total order that satisfies at least $m(1-\varepsilon)$ of the constraints (even if they are all satisfiable) is NP-hard. In particular this holds for every $\varepsilon < 1/48$ (see Corollary 2.5). On the other hand, it is easy to find a total order that satisfies $1/3$ of the m constraints (even if they are not all satisfiable). Simply arrange the points in a random order along the line. The probability that a specific constraint (x_i, x_j, x_k) is satisfied by such a randomly chosen order is $1/3$, since exactly two of the six permutations on i, j, k have j in the middle. Thus the expected number of constraints satisfied by a random order is at least $1/3$ of the m constraints. On the other hand, it is easy to construct examples where at most $m/3$ constraints are satisfiable. Thus to achieve better approximation factors, one needs to be able to recognize instances of the betweenness problems that are not satisfiable.

We present a polynomial time algorithm that either determines that there is no feasible solution or finds a total order that satisfies at least $1/2$ of the m constraints. Our algorithm translates the problem into a set of quadratic inequalities and solves a semidefinite programming (SDP) relaxation of them in $\mathcal{R}^n$. Let $v_1, \ldots, v_n \in \mathcal{R}^n$ be a feasible solution to the SDP, where each v_i corresponds to the real variable x_i. The n solution points are then projected on a random line through the origin. We show that if "x_j between x_i and x_k" is one of the betweenness constraints, then the angle between the lines v_iv_j and v_kv_j (in $\mathcal{R}^n$) is obtuse. Using this property, we prove that the random projection satisfies each constraint with probability at least $1/2$. This gives a randomized algorithm with the claimed performance guarantee. Next, we show how to derandomize the algorithm. In addition, we demonstrate that our analysis of the semidefinite program is tight. There is an infinite family of inputs to the betweenness problem, such that the resulting SDP is feasible, but any total order of the variables satisfies at most $1/2 + o(1)$ of the m constraints.

Our use of semidefinite programming is inspired by the recent success in using this methodology to find improved approximation algorithms for several optimization problems. The applicability of SDP in combinatorial optimization was demonstrated by Grötschel, Lovász, and Schrijver [7] to show that the Theta function of Lovász [12] was polynomial time computable. This application was then turned into exact coloring and independent set finding algorithms for perfect graphs. The use of SDP in approximation algorithms was innovated by the work of Goemans and Williamson [5] who broke longstanding barriers in the approximability of MAX CUT and MAX 2SAT by their SDP based algorithm. Further evidence of the applicability of the SDP approach is provided by the works of Karger, Motwani, and Sudan [10], who use it to approximate graph coloring, Alon and Kahale [1] (independent set approximation), and Feige and Goemans [4] (improvements to MAX 2SAT).

Thus the semidefinite programming method has now been used successfully to solve many optimization problems—exactly and approximately. However, all the cases where SDP has been used to find approximation algorithms seem to be essentially partition problems (MAX CUT, Coloring, Multicut, etc.). Our solution seems to be (to the best of our knowledge) the only case where SDP has been used to solve an

ordering problem. This syntactic difference between ordered structures and unordered ones, and the ability of SDP to help optimize over both, offers critical additional evidence on the power of the SDP methodology.

The remainder of this paper is organized as follows. Section 2 presents the approximation preserving reduction from MAX 3SAT, as well as other observations about the betweenness problem. Semidefinite programming is briefly reviewed in section 3. The algorithm is presented in section 4. Section 5 shows the tightness of our analysis. Finally, section 6 contains some concluding remarks and open problems.

2. Preliminaries. We start this section with some preliminary observations about the betweenness problem. We begin by defining the notion of an approximate solution to the betweenness problem and analyzing the complexity of finding such a solution.

DEFINITION 2.1. *Given an instance of the betweenness problem on m constraints and $\alpha \leq 1$, an* α-approximate solution *is one that satisfies at least αk constraints, where k is the maximum number of constraints satisfied by any solution. For $\alpha \leq 1$, the α-approximation (version of the betweenness) problem is the task of finding an α-approximate solution for every instance. An algorithm that solves such a problem is said to be an α-approximation algorithm. For $\alpha \leq 1$, the α-approximation problem for* satisfiable *instances is the task of finding a total order that satisfies αm constraints or determining that the instance is not satisfiable. An algorithm that solves this problem is an α-approximation algorithm for satisfiable instances.*

The complexity of solving the betweenness problem exactly (i.e., for $\alpha = 1$) is well settled. Opatrny [14] has shown that it is NP-hard to decide if a given instance of the betweenness problem is satisfiable. We now turn our attention to the complexity of the problem for $\alpha < 1$. We first present a hardness result based on a simple reduction from MAX CUT, due to Goemans (personal communication, 1995). An instance of the MAX CUT problem is an undirected graph. The goal of the problem is to find a partition $(S, \bar{S})$ of the vertex set so as to maximize the number of edges with one endpoint in S and one in $\bar{S}$. This problem was shown by Arora et al. [2] to be hard to approximate to within some factor $\alpha < 1$. The best result known to date, due to Håstad [9] (see also Trevisan et al. [17]), is that α-approximating MAX CUT is NP-hard for every $\alpha > 16/17$.

PROPOSITION 2.2. *For every α, the α-approximation version of the MAX CUT problem reduces to the α-approximation version of the betweenness problem.*

Proof. Given an instance G of the MAX CUT problem, we create an instance of the betweenness problem as follows: For every vertex v_i in the graph, create a point p_i. In addition we introduce one special point s. For every edge (v_i, v_j) in the graph, we introduce the betweenness constraint (p_i, s, p_j) (i.e., s is between p_i and p_j). Now, given a cut $(S, \bar{S})$ in the graph that has k edges crossing the cut, any ordering that places the points corresponding to the vertices in S to the left of s and the rest of the points to the right of s is an ordering that satisfies k of the betweenness constraints. In the reverse direction, any ordering of the points that satisfies k betweenness constraints can be converted into a cut with k edges crossing the cut by letting S be the set of vertices corresponding to points to the left of s. Thus the optima of the two problems are exactly equal; furthermore, given an α-approximate solution to the betweenness instance, we can construct an α-approximate solution to the MAX CUT instance. Thus an α-approximation algorithm for the betweenness problem yields an α-approximation algorithm for the MAX CUT problem. ∎

COROLLARY 2.3. *The α-approximation version of the betweenness problem is*

NP-hard for $\alpha > 16/17$.

While the above reduction provides some insight about the hardness of the betweenness problem on general instances, it does not quite provide a hardness result for the problem of interest to us. This is because the instances of the betweenness problem that we typically consider are fully satisfiable. In the reduction above, the only instances of the MAX CUT problem that reduce to fully satisfiable instances of betweenness are when the input graph is bipartite. But in such cases it is easy to find the MAX CUT, and thus the instances of betweenness produced are not necessarily hard.

In what follows we present an approximation preserving reduction from MAX 3SAT to the betweenness problem. This reduction follows Opatrny's original reduction and addresses the α-approximation problem for satisfiable instances. It is well known that there exists a constant $\varepsilon > 0$ such that the $(1-\varepsilon)$-approximation version of the MAX 3SAT problem is NP-hard. The best results known to date, due to Håstad [9], show that this is true for every $\varepsilon < 1/8$. Based on our reduction we conclude that there exists a constant $\varepsilon' > 0$ such that finding an ordering that satisfies a $(1-\varepsilon')$ fraction of the constraints in a *satisfiable* instance of the betweenness problem is NP-hard.

PROPOSITION 2.4. *For every* $\varepsilon > 0$, *the* $(1-\varepsilon)$*-approximation version of the MAX* 3*SAT problem on satisfiable instances reduces to the* $(1-\varepsilon/6)$*-approximation version of the betweenness problem on satisfiable instances.*

Proof. Given a 3-CNF formula ϕ on n variables and m clauses, we construct an instance I of the betweenness problem on $2+n+5m$ points with $6m$ constraints such that, for every ℓ, there exists a total order satisfying $5m+\ell$ of the betweenness constraints in I if and only if there exists an assignment satisfying ℓ of the clauses in ϕ. The reduction proceeds as follows: For each Boolean variable x_i of ϕ, we add a point p_i to I. In addition we create two special points T and F. Without loss of generality, we consider orderings where T is to the right of F. An ordering of the points p_i, T, and F is supposed to imply a truth assignment as follows: If p_i is to the left of F then it is false; if it is to the right of F then it is true. This interpretation will also apply to the additional "clause points" that are introduced in the rest of the construction.

Given a clause C_j, say $C_j = x_1 \vee \overline{x_2} \vee x_3$, we create five points $q_j^{(1)}$, $q_j^{(\bar{2})}$, and $q_j^{(3)}$ and $r_j^{(12)}$ and $r_j^{(123)}$. The points q_j are supposed to represent the assignment to the literals in the clause. For each literal in the clause, we include a constraint that forces the variable to be assigned consistently with the literal. We do so with the following constraints: F between p_2 and $q_j^{(\bar{2})}$, whereas $q_j^{(1)}$ is between p_1 and F, and $q_j^{(3)}$ is between p_3 and F. Thus for example, an assignment satisfies $q_j^{(\bar{2})}$ if and only if it falsifies p_2. The points $r_j^{(12)}$ and $r_j^{(123)}$ are supposed to represent the OR of the first two and three literals in the clause, respectively. This is enforced with the following betweenness constraints: $r_j^{(12)}$ is between $q_j^{(1)}$ and $q_j^{(\bar{2})}$ and $r_j^{(123)}$ is between $r_j^{(12)}$ and $q_j^{(3)}$. So, for example, if both literal points $q_j^{(1)}$ and $q_j^{(\bar{2})}$ are false, and $r_j^{(12)}$ is between $q_j^{(1)}$ and $q_j^{(\bar{2})}$, then $r_j^{(12)}$ must be false, while if at least one of the literal points is true, then $r_j^{(12)}$ can be placed so that it is true (to the right side of F). Lastly we add a betweenness constraint that attempts to ensure that a clause is assigned true. This is done with the following constraint: $r_j^{(123)}$ is between F and T.

Thus corresponding to each clause we have six betweenness constraints. Consider an assignment to the variables in ϕ satisfying ℓ clauses out of m. Without loss of

generality, assume that the assignment sets $x_1, \ldots, x_k = \mathsf{false}$ and $x_{k+1}, \ldots, x_n = \mathsf{true}$. Order the points p_i and T and F as follows:

$$p_1 \cdots p_k \; F \; p_{k+1} \cdots p_n \; T.$$

For j going from 1 to m, the literal points $q_j^{(1)}$, $q_j^{(2)}$, and $q_j^{(3)}$ are then placed between p_k and F or between F and p_{k+1}, depending on their truth value. (A true literal is placed between F and p_{k+1} while a false literal is between p_k and F.) Finally, the points $r_j^{(12)}$ and $r_j^{(123)}$ are placed as far to the right as possible subject to the betweenness constraints. This tends to make $r_j^{(123)}$ lie between F and T if any one of the literals in the jth clause is true. This arrangement always satisfies at least five of the betweenness constraints associated with the kth clause. The only constraint it may not satisfy is the constraint "$r_j^{(123)}$ is between F and T"; this constraint is satisfied if and only if at least one of the literals in the jth clause is true. Thus this ordering satisfies $5m + \ell$ of the betweenness constraints. Conversely it may be verified that if an arrangement of the points (again, with F left of T) satisfies $5m + \ell$ betweenness constraints, then the assignment that assigns true to all of those variables whose corresponding points lie to the right of F satisfies at least ℓ clauses in the formula ϕ. (There must be at least ℓ values of j for which the arrangement satisfies all six betweenness constraints involving q_j's and r_j's. For these values of j, the corresponding assignment satisfies the jth clause.)

Thus given a 3-CNF formula ϕ with m clauses, we have constructed a betweenness instance I with $m' = 6m$ constraints. Furthermore, given an ordering satisfying $(1-\varepsilon)m'$ constraints, we can reconstruct an assignment satisfying at least $(1-\varepsilon)m' - 5m = m(1-6\varepsilon)$ clauses of ϕ. □

COROLLARY 2.5. *The α-approximation version of the betweenness problem on satisfiable instances is NP-hard, for every $\alpha > 47/48$.*

Next we show what can achieved by the obvious randomized algorithm for the betweenness problem.

The natural randomized algorithm for the betweenness problem arranges the points in a random order along the line. The probability that a specific constraint is satisfied by such a randomly chosen order is 1/3. Thus the expected number of constraints satisfied by a random order is at least 1/3 of all the constraints. By the method of conditional probabilities one can find such order in polynomial time. Since this order satisfies 1/3 of all constraints, it is within 1/3 of the optimal ordering. The result is summarized below.

PROPOSITION 2.6. *The 1/3-approximation version of the betweenness problem can be solved in polynomial time.*

Before going on to more sophisticated techniques for solving this problem, let us examine the main weakness of the above algorithm. We first argue that no algorithm can do better than attempting to satisfy 1/3 of all given constraints. Consider an instance of the betweenness problem on three points with three constraints insisting that each point be between the other two. Clearly we can satisfy only one of the above three constraints, which proves the claim. Thus the primary weakness of the above algorithm is not in the (absolute) number of constraints it satisfies, but in the fact that it attempts to do so for every instance of the betweenness problem—even those that are obviously not satisfiable. Thus to achieve better approximation factors, one needs to be able to recognize instances of the betweenness problems that are not satisfiable. However, this is an NP-hard task. In fact, Corollary 2.5 indicates

that one cannot even distinguish instances that are satisfiable from those for which an ε fraction of the constraints remain unsatisfied under any assignment. In what follows we use a semidefinite relaxation of our problem to distinguish cases that are not satisfiable from cases where at least 50% of the given clauses are satisfiable. We then go on to show that using this relaxation we can achieve a better approximation than the naive randomized algorithm.

3. Semidefinite programming (SDP). In this section we briefly introduce the paradigm of SDP. We describe why it is solvable in polynomial time. A complementary technique to that of SDP is the incomplete Cholesky decomposition. We describe how the combination allows one to find embeddings of points in finite-dimensional Euclidean space, subject to certain constraints.

DEFINITION 3.1. *For positive integers m and n, a semidefinite program is defined over a collection of n^2 real variables $\{x_{ij}\}_{i=1,j=1}^{n,n}$. The input consists of a set of mn^2 real numbers $\{a_{ij}^{(k)}\}_{i=1,j=1,k=1}^{n,n,m}$, a vector of m real numbers $\{b^{(k)}\}_{k=1}^{m}$, and a vector of n^2 real numbers $\{c_{ij}\}_{i=1,j=1}^{n,n}$. The objective is to find $\{x_{ij}\}_{i=1,j=1}^{n,n}$ so as to*

$$\begin{aligned} &\textit{maximize} && \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ &\textit{subject to} && \\ \forall k \in \{1,\ldots,m\} \quad & && \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^{(k)}x_{ij} \le b^{(k)} \\ &\textit{and} && \textit{the matrix } X = \{x_{ij}\} \textit{ is symmetric} \\ & && \textit{and positive semidefinite.} \end{aligned}$$

Recall that the following are equivalent ways of defining when a symmetric matrix X is positive semidefinite.

1. All the eigenvalues of X are nonnegative.
2. For all vectors $y \in \mathcal{R}^n$, $y^T X y \ge 0$.
3. There exists a real matrix V such that $V^T \cdot V = X$.

It is well known that the ellipsoid algorithm of Khaciyan [11] can be used to solve any semidefinite program approximately in the following sense: Given a parameter $\varepsilon > 0$, the algorithm runs in time polynomial in the input size and $\log(1/\varepsilon)$ and finds a feasible solution achieving an objective of at least optimum $-\varepsilon$ (see, for instance, [8]).

In order to use the semidefinite programming approach for solving combinatorial optimization problems, one more tool is useful. This is the ability to find a matrix V as guaranteed to exist in part 3 of the above definition of positive semidefiniteness. The method that yields such a matrix is the incomplete Cholesky decomposition.

The matrix V can be used to interpret the solution obtained by the SDP problem geometrically. Interpret the columns of the $n \times n$ matrix V as n vectors $v_1, \ldots, v_n$ in $\mathcal{R}^n$. Now the variables x_{ij} of the matrix X correspond simply to the inner product of v_i and v_j. Thus a linear constraint on the x_{ij}'s is simply a linear constraint on the inner products of the v_i's vectors. Also, the objective function is simply a linear function on the inner products.

Thus the following provides an equivalent geometric interpretation of SDP:

> Find n vectors $v_1, \ldots, v_n$ so as to maximize the quantity $\sum_{i,j} c_{ij}\langle v_i, v_j\rangle$, subject to the constraints $\sum_{i,j} a_{ij}^{(k)}\langle v_i, v_j\rangle \le b^{(k)}$, for every $k \in \{1,\ldots,m\}$.

Alternately one can interpret SDP as solving an optimization problem that attempts to find n points in n-dimensional Euclidean space, subject to linear constraints on the squares of the distance between the points. This is done by observing that the square of the distance between points v_i and v_j (denoted d_{ij}^2) is simply

$$\langle (v_i - v_j), (v_i - v_j) \rangle = \langle v_i, v_i \rangle + \langle v_j, v_j \rangle - 2\langle v_i, v_j \rangle.$$

Thus a linear inequality on the d_{ij}^2's is also a linear inequality on the inner products of the v_i's. (Actually the distance squared interpretation is equivalent to SDP since we can express $\langle v_i, v_j \rangle$ as $(d_{i0}^2 + d_{j0}^2 - d_{ij}^2)/2$.)

From this interpretation of SDP we can solve any problem of the form:

> **Geometric SDP.** Embed n points in $\mathcal{R}^n$ such that the squares of the distance between the points, denoted d_{ij}, satisfy the constraints $\sum_{i,j} a_{ij}^{(k)} d_{ij}^2 \leq b^{(k)}$ while trying to maximize $\sum_{i,j} c_{ij} d_{ij}^2$. In the *$\varepsilon$-additive approximation* version the algorithm is allowed to return (for every feasible input) a solution such that each constraint is violated by at most ε, i.e., $\sum_{i,j} a_{ij}^{(k)} d_{ij}^2 \leq b^{(k)} + \varepsilon$, and the objective achieved is at least the optimum $-\varepsilon$.

In what follows we will use the last interpretation of SDP to solve the betweenness problem. In particular, we use the following proposition.

PROPOSITION 3.2. *For every $\varepsilon > 0$, the ε-additive approximation version of the geometric SDP can be solved in time polynomial in the input size and* $\log(1/\varepsilon)$.

4. The algorithm. The general idea of our algorithm is to express the betweenness constraints as a set of real quadratic inequalities. By considering an n-dimensional relaxation of the problem, we get an instance of SDP and can find a feasible solution in $\mathcal{R}^n$ (if one exists). We study simple geometric properties of this solution set. We use them to argue that a projection of the set on a random line satisfies at least $1/2$ of the betweenness constraints (with high probability). Then we show how to derandomize the algorithm.

Consider a set of m betweenness constraints on n real variables $x_1, \ldots, x_n$. Suppose these constraints are satisfiable and that $x_1 < x_2 < \cdots < x_n$ is a satisfying linear order. We can clearly embed the points in the unit interval and assign $x_i = (i-1)/(n-1)$ $(i = 1, \ldots, n)$. Let x_i, x_j, x_k be a triplet such that x_j is required to be between x_i and x_k. For the assignment above, it is readily seen that $(x_i - x_j)^2 + (x_k - x_j)^2 < (x_i - x_k)^2$. Furthermore, the x's are at least $1/(n-1)$ apart and at most 1 apart. Thus for every pair of distinct indices i, j, the x's satisfy the inequalities $1/(n-1)^2 \leq (x_i - x_j)^2 \leq 1$. This motivates the following geometric SDP relaxation for the betweenness problem.

(SDP1(I)) Embed n points in $\mathcal{R}^n$ subject to the constraints

$$\frac{1}{(n-1)^2} \leq d_{ij}^2 \leq 1 \qquad \forall\, i \neq j,$$
$$d_{ij}^2 + d_{jk}^2 \leq d_{ik}^2 \qquad \text{for every constraint } (x_i, x_j, x_k).$$

We strengthen this relaxation slightly before showing how to use it to find an approximate solution to the instance of the betweenness problem. Recall that the x's are at least $1/(n-1)$ apart and at most 1 apart. Therefore for any triple (x_i, x_j, x_k), the ratio between $(x_i - x_j)^2 + (x_j - x_k)^2$ and $(x_i - x_k)^2$ is maximized when x_i and x_k are extreme points (0 and 1), and x_j is as close as possible to one of them $(1/(n-1)$

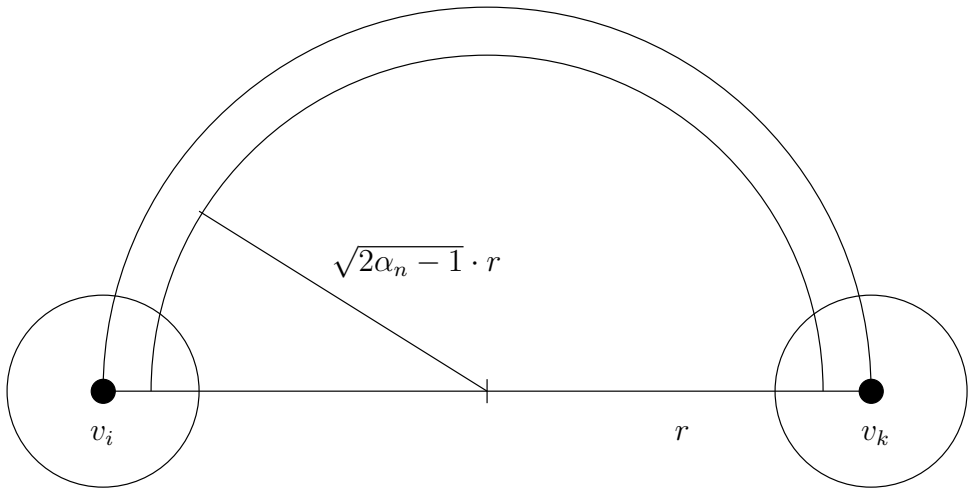


FIG. 4.1. *Possible location for the midpoint v_j.*

or $(n-2)/(n-1)$). For these values, the ratio is

$$\left(\frac{1}{n-1}\right)^2 + \left(1 - \frac{1}{n-1}\right)^2 = 1 - \frac{2}{n-1} + \frac{2}{(n-1)^2}.$$

Denote this value by α_n. Notice that $\alpha_n = 1 - 2/n + o(1/n)$ depends only on the number of variables.

We are now ready to set up our final SDP relaxation:

(SDP(I)) Embed n points in $\mathcal{R}^n$ subject to the constraints

$$\frac{1}{(n-1)^2} \le d_{ij}^2 \le 1 \qquad \forall\, i \ne j,$$
$$d_{ij}^2 + d_{jk}^2 \le \alpha_n d_{ik}^2 \qquad \text{for every constraint } (x_i, x_j, x_k).$$

The argument leading to the construction of the instance SDP(I) says that the SDP is feasible if the instance I is satisfiable and in fact there exists an embedding of the points in one dimension satisfying all the constraints. We summarize this below.

PROPOSITION 4.1. *For every instance I of the betweenness problem, if I is satisfiable, then the semidefinite program SDP(I) is feasible.*

As argued in section 3 (see Proposition 3.2), we can use the ellipsoid algorithm to test the feasibility of SDP(I) and, if it is feasible, to find an approximation of a feasible solution (if one exists). Let $v_1, \ldots, v_n \in \mathcal{R}^n$ be an approximately feasible solution, and let $v_i, v_j, v_k \in \mathcal{R}^n$ be a triplet that corresponds to a betweenness constraint. We first prove some geometric facts about the points v_i, v_j, v_k and then use this to design our approximation algorithm.

Consider any two-dimensional plane through the points v_i, v_j, v_k. (If v_i, v_j, v_k are not collinear, then this plane is unique; otherwise we pick any such plane arbitrarily.) Let $2r$ be the distance between v_i and v_k $(1/(n-1) - \varepsilon \le 2r \le 1 + \varepsilon)$. In what follows we shall skip the term ε since it can be made arbitrarily small (and, in particular, exponentially small in n).

We now consider the angle $\theta_{i,j,k} = \angle v_i v_j v_k$. We claim that this angle is obtuse (i.e., at least $\pi/2$). To see this, we project the points down to the two-dimensional plane containing v_i, v_j, and v_k. Furthermore, we rotate and translate the points so

that $v_i = (-r, 0)$, $v_k = (r, 0)$, and $v_j = (x, y)$. Now we can use the explicit formulae $d_{ij}^2 = (x + r)^2 + y^2$, $d_{jk}^2 = (r - x)^2 + y^2$ and $d_{ik}^2 = 4r^2$. The constraint on these distances yields

$$(x - r)^2 + y^2 + (x + r)^2 + y^2 \leq 4\alpha_n r^2,$$

which implies

$$x^2 + y^2 \leq (2\alpha_n - 1)r^2.$$

This means that v_j, the "midpoint" in the betweenness constraint, lies inside a ball of radius $r\sqrt{2\alpha_n - 1}$, whose center is the middle point $(v_i + v_k)/2$, and outside the two small balls of radius $1/(n-1)$ around v_i and v_k (see Figure 4.1).

This proves that the angle $\theta_{i,j,k} = \angle v_i v_j v_k$ is indeed obtuse. The following claim proves a tighter bound on $\theta_{i,j,k}$.

CLAIM 4.2. *The angle $\theta_{i,j,k}$ satisfies $\theta_{i,j,k} \geq (1 + \Omega(1/n))\pi/2$.*

Proof. We apply the cosine rule

$$\begin{aligned}
\cos\theta_{i,j,k} &= (d_{ij}^2 + d_{jk}^2 - d_{ik}^2)/(2d_{ij}d_{jk}) \\
&= (x^2 + y^2 - r^2)/\left(\sqrt{(x^2 + y^2 + r^2)^2 - 4r^2x^2}\right) \\
&\leq (x^2 + y^2 - r^2)/(x^2 + y^2 + r^2) \\
&\leq (\alpha_n - 1)/\alpha_n \\
&< \alpha_n - 1 \\
&= -\frac{2}{n} + \theta\left(\frac{1}{n^2}\right) .
\end{aligned}$$

Denoting $\theta_{i,j,k} = h + \pi/2$ and using the Taylor series expansion

$$\cos(h + \pi/2) = -h + \frac{h^3}{6} - \frac{h^5}{120} + \cdots,$$

we get

$$-h + \theta(h^3) \leq -\frac{2}{n} + \theta\left(\frac{1}{n^2}\right)$$

so $h = \Omega(\frac{1}{n})$, namely, $\theta_{i,j,k} \geq (1 + \Omega(1/n))\pi/2$. ∎

We are now ready to describe our algorithm. The algorithm proceeds by picking uniformly at random a line through the origin and projecting the n points $v_1, \ldots, v_n$ on this random line. Let $x'_1, \ldots, x'_n$ be the n resulting points.

CLAIM 4.3. *Let $\theta_{i,j,k}$ denote the angle $\angle v_i v_j v_k$. Then the probability that x'_j lies between x'_i and x'_k equals $\theta_{i,j,k}/\pi$.*

Proof. Instead of considering an arbitrary line through the origin, we consider a parallel line that goes through the point v_j. This does not change the betweenness relation of the projections; neither is this relation changed when considering the projection of this line on the two-dimensional plane defined by v_i, v_j, v_k. Consider the section of the circle defined by the two lines that go through v_j and are perpendicular to the lines $v_i v_j$ and $v_k v_j$. It is not hard to see that only lines going through this section violate the betweenness constraint of the projections. This section occupies an angle of $\pi - \theta_{i,j,k}$ (see Figure 4.2). The claim follows. ∎

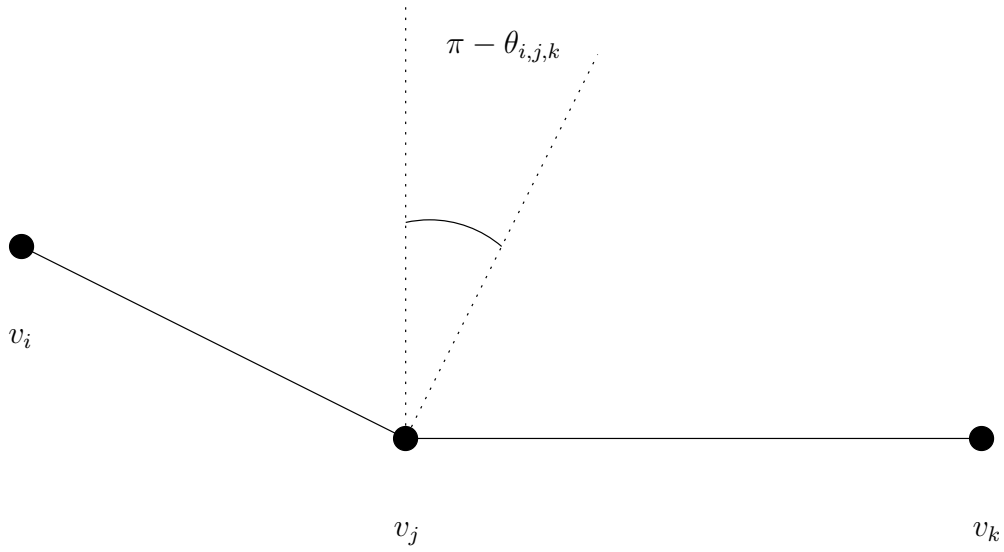


FIG. 4.2. *Lines going through the circular section violate the constraint.*

Combining Claims 4.2 and 4.3, we get the following.

COROLLARY 4.4. *Suppose SDP(I) has a feasible solution. Then for any of the m constraints, the probability that x'_j lies between x'_i and x'_k is at least $1/2 + \Omega(1/n)$.*

As a consequence, the expected number of betweenness constraints satisfied by $x'_1, \ldots, x'_n$ is at least $m/2 + \Omega(m/n) = m(1/2 + \Omega(1/n))$. This yields the following lemma that forms a (weak) converse to Proposition 4.1.

LEMMA 4.5. *For any instance I of the betweenness problem, if SDP(I) is feasible, then there exists a total order satisfying at least $m/2 + \Omega(m/n)$ of the betweenness constraints in I.*

Thus we get a randomized polynomial time algorithm that either finds that the constraints are infeasible or generates a linear order that satisfies at least $1/2$ the constraints.

We now outline a method for derandomizing our algorithm. Given an embedding of the betweenness problem, we can define a graph and an embedding of the graph in $\mathcal{R}^n$ such that the expected size of the MAX CUT found for this embedding of the graph equals the expected number of betweenness constraints that are satisfied by a random projection.

For every ordered pair of points (v_i, v_j) of the betweenness problem, introduce the vertex w_{ij} with embedding $v_i - v_j$. If i, j, k is a betweenness constraint, then put an edge between w_{ij} and w_{kj}. This defines the graph and its embedding.

Now consider any hyperplane through the origin that cuts across the edge between w_{ij} and w_{kj}. Let the slope of the normal to the hyperplane be the vector r. Assume, without loss of generality, that $r.w_{ij} < 0$ and $r.w_{jk} < 0$, then $r.v_i < r.v_j$ and $r.v_j < r.v_k$. Thus j lies between i and k. Conversely, if projection onto the vector r satisfies the betweenness constraint for i, j, k, then the edge between w_{ij} and w_{jk} must be cut.

Mahajan and Ramesh [13] give a method to deterministically find a vector r whose cut value equals the expected cut value. They use this algorithm to derandomize the MAX CUT and MAX 2SAT algorithm of Goemans and Williamson [5]. By using their algorithm, we get a vector such that projection onto this vector satisfies as many

constraints as the expected number satisfied by a random vector.

Remark. Observe that the above reduction is not a generic reduction from betweenness to MAX CUT. It uses the fact that the graph produced for the MAX CUT problem has a specified embedding in order to map a solution of the MAX CUT problem to a solution of the betweenness problem.

We conclude this section by stating the main theorem of this paper.

THEOREM 4.6. *The* $1/2$*-approximation version of the betweenness problem can be solved in polynomial time. Specifically, there exists a polynomial time algorithm which takes as input an instance of the betweenness algorithm on* n *points and* m *constraints and either outputs "not feasible" or outputs a total order satisfying at least* $m/2 + \Omega(m/n)$ *constraints.*

5. Tightness of our analysis. In this section we show that our analysis of the semidefinite program is almost tight. We do so by exhibiting two families of instances of the betweenness problem on m constraints, such that the optimum value is at most $m(1/2 + o(1))$, but (a slight perturbation of) the SDP is nevertheless feasible.

The first example is related to the d-dimensional hypercube. For every integer $d > 1$, we construct the instance I_d as follows. I_d has 2^d points corresponding to the 2^d vertices of the d-dimensional hypercube. I_d has $m = \binom{d}{2}2^d$ constraints—one for every simple path of length 2 in the hypercube, with the betweenness constraint expecting the middle vertex of the path to be between the endpoints.

Consider a small perturbation of our SDP, where we set $d_{i,j}^2 + d_{j,k}^2 \leq d_{i,k}^2$ for each betweenness constraint. This SDP is clearly feasible—the natural embedding of the hypercube in d-dimensions (as a hypercube) ensures that every path of length 2 subtends an angle of 90° at their midpoint.

Now consider a linear ordering of the points. Consider any point p and all the paths that have p as their midpoint. The number of such paths is $\binom{d}{2}$. Now let d_1 of the neighbors of p be on its left and d_2 of its neighbors be on its right (where $d_1 + d_2 = d$). The number of betweenness constraints expecting p to be in the middle that get satisfied is $d_1 d_2 \leq d^2/4$. Thus, for any point, the fraction of betweenness constraints that are associated with the point and are satisfied is at most $(d^2/4)/(d(d-1)/2) = d/(2(d-1)) = 1/2 + 1/(2(d-1)) = 1/2 + o(1)$.

The second example, suggested to us by Goemans, is related to the cuts in the complete graph K_n on n variables. For every integer $n > 1$, we construct the instance C_n as follows. C_n has $n+1$ points, a "center point" v_0 and n "vertices" $v_1, \ldots, v_n$. C_n has $m = \binom{n}{2}$ constraints—one for every edge in the complete graph. For every $1 \leq i < k \leq n$, we have the betweenness constraint that v_0 is between v_i and v_k.

We now consider the following perturbation of our SDP, where $d_{i,j}^2 + d_{j,k}^2 \leq (1-1/n)d_{i,k}^2$ for each betweenness constraint. To see that this SDP is feasible, consider the following embedding: The vertex v_i is embedded as the point $(0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 occurs in the ith coordinate. The vertex v_0 is embedded as the point $(1/n, \ldots, 1/n)$. Observe that the distance between v_i and v_j is $\sqrt{2}$ and the distance between v_i and v_0 is $\sqrt{1-1/n}$. Thus for any two indices $i, k \neq 0$ the inequality $d_{i,0}^2 + d_{0,k}^2 \leq (1 - 1/n)d_{i,k}^2$, which corresponds to the betweenness constraints, is satisfied (in fact, equality holds). Now in order to satisfy the SDP (recall that we required all pairwise distances to be at most 1) we simply scale down the simplex so that the distance between the vertices is 1, embed the center v_0 in the origin and each vertex v_i in the corresponding simplex vertex. This embedding satisfies all of the SDP constraints.

Again, any linear ordering of the $n+1$ points induces a cut in the graph K_n

(vertices to the left of v_0; vertices to the right of v_0). An edge corresponds to a satisfied betweenness constraint if and only if the edge is across the cut. Therefore the maximum number of satisfiable constraints equals the sized of a maximum cut in K_n, namely, $(n/2)^2 = m(1/2 + o(1))$.

The advantage of this maximum cut example is that it shows tightness of the analysis with respect to quadratic inequalities of the form

$$d_{i,j}^2 + d_{k,j}^2 \leq \beta_n d_{i,k}^2,$$

where $\beta_n = 1 - 1/n - o(1/n)$. Our original SDP has the form

$$d_{i,j}^2 + d_{k,j}^2 \leq \alpha_n d_{i,k}^2,$$

where $\alpha_n = 1 - 2/n + o(1/n)$. By starting with the complete graph example, and padding it with extra dummy variables that do not take part in any constraint, we can construct an example where only $1/2 + o(1)$ of the constraints are satisfiable, yet the original SDP (with α_n) is feasible (in fact any $\gamma_n = o(1)$ can work here). It is not clear how to come up with a nonartificial construction, i.e., without padding, having these properties.

6. Concluding remarks. We remark that metric information can be easily incorporated into our algorithm. As a simple example, suppose that for some of the constraints we know not only that x_j is between x_i and x_k, but that it is exactly in the middle, namely, $x_j = (x_i + x_k)/2$. In this case, we add the inequality

$$d_{i,j}^2 + d_{k,j}^2 \leq d_{i,k}^2/4$$

instead of

$$d_{i,j}^2 + d_{k,j}^2 \leq \alpha_n d_{i,k}^2.$$

Any feasible solution will have v_j exactly in the middle of v_i and v_k, and the same holds with respect to the final projections.

Finally, notice that our formulation of the problem as SDP tested only for feasibility of the constraints. It is interesting to see if the inclusion of an appropriate objective function, and possibly of additional inequalities, can be used to improve the performance guarantee of the algorithm. Other approaches to the problem, possibly purely combinatorial ones, are also of interest.

Acknowledgments. We are grateful to Michel Goemans for providing us with the MAX CUT example and for helpful discussions on semidefinite programming. Many thanks to Amir Ben-Dor for numerous helpful discussions on the betweenness problem. We would also like to thank Ron Shamir for acquainting us with reference [14] and for useful discussions; Oded Goldreich and the anonymous referee for their comments on earlier versions of this paper; and Amos Beimel and Dan Pelleg for their expert advice on xfig.

REFERENCES

[1] N. Alon and N. Kahale, *Approximating the independence number via the θ-function*, Math. Programming, Ser. A, 80 (1998), pp. 253–264.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and hardness of approximation problems*, J. Assoc. Comput. Mach., to appear. An extended abstract appears in Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1992, pp. 14–23.

[3] D. Cox, M. Burmeister, E. Price, S. Kim, and R. Myers, *Radiation hybrid mapping: A somatic cell genetic method for constructing high resolution maps of mammalian chromosomes*, Science, 250 (1990), pp. 245–250.

[4] U. Feige and M. Goemans, *Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT*, in Proc. Third Israel Symposium on Theory and Computing Systems, Tel Aviv, Israel, 1995, IEEE Computer Society Press, Los Alamitos, CA, pp. 182–189.

[5] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42(1995), pp. 1115–1145.

[6] S. Goss and H. Harris, *New methods for mapping genes in human chromosomes*, Nature, 255 (1975), pp. 680–684.

[7] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[8] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization,* Springer-Verlag, Berlin, 1987.

[9] J. Håstad, *Some optimal inapproximability results*, in Proc. Twenty-Ninth Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997, ACM, New York, pp. 1–10.

[10] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring via semidefinite programming*, J. Assoc. Comput. Mach., to appear. An extended abstract appears in Proc. 35th Annual IEEE Symposium on Foundations of Computer Science, Santa Fe, NM, 1994, IEEE Computer Society Press, Los Alamitos, CA, pp. 2-13.

[11] L. Khaciyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194 (English translation).

[12] L. Lovász, *On the Shannon capacity of a graph*, IEEE Trans. Inform. Theory, IT-25 (1979), pp. 1–7.

[13] S. Mahajan and H. Ramesh, *Derandomizing semidefinite programming based approximation algorithms*, in Proc. 36th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 162–169.

[14] J. Opatrny, *Total ordering problem*, SIAM J. Comput., 8 (1979), pp. 111–114.

[15] D. Slonim, L. Stein, L. Kruglyak, and E. Lander, *RHMAPPER: An interactive computer package for constructing radiation hybrids maps*, 1996. Available online at http://www.genome.wi.mit.edu/ftp/pub/software/rhmapper.

[16] D. Slonim, L. Stein, L. Kruglyak, and E. Lander, *Building human genome maps with radiation hybrids*, J. Comput. Biol., 4 (1997), pp. 487–504.

[17] L. Trevisan, G. Sorkin, M. Sudan, and D. Williamson, *Gadgets, approximation, and linear programming*, in Proc. 37th Annual IEEE Symposium on Foundations of Computer Science, Burlington, VT, 1996, IEEE Computer Society Press, Los Alamitos, CA, pp. 617–626.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 524–545, November 1998

002

THE GRAPHS WITH ALL SUBGRAPHS T-PERFECT*

A. M. H. GERARDS† AND F. B. SHEPHERD‡

Abstract. The richest class of t-perfect graphs known so far consists of the graphs with no so-called odd-K_4. Clearly, these graphs have the special property that they are *hereditary t-perfect* in the sense that every subgraph is also t-perfect, but they are not the only ones. In this paper we characterize hereditary t-perfect graphs by showing that any non–t-perfect graph contains a non–t-perfect subdivision of K_4, called a *bad-K_4*. To prove the result we show which "weakly 3-connected" graphs contain no bad-K_4; as a side-product of this we get a polynomial time recognition algorithm.

It should be noted that our result does not characterize t-perfection, as that is not maintained when taking subgraphs but only when taking induced subgraphs.

AMS subject classifications. 05C75, 05C70, 90C10, 90C27

Key words. stable sets, polyhedra, odd circuits, decomposition

PII. S0895480196306361

1. Introduction.

A graph $G = (V, E)$ is *t-perfect* if the polyhedron

$$\mathcal{P}(G) := \{x \in \mathbb{R}^V \mid \begin{array}{rcll} x_v & \geq & 0 & (v \in V), \\ x_u + x_v & \leq & 1 & (uv \in E), \\ \sum_{v \in V(C)} x_v & \leq & \frac{|V(C)|-1}{2} & (C \text{ is odd circuit in } G) \end{array} \tag{1}$$

has integral vertices only, i.e., when $\mathcal{P}(G)$ is the stable set polytope of G. T-perfection was introduced by Chvátal [4], and a characterization of it has proved elusive. The first two classes of graphs known to be t-perfect are series-parallel graphs (conjectured by Chvátal [4] and proved by Boulala and Uhry [2]) and *almost bipartite graphs*, i.e., graphs with a node that is contained in every odd circuit [5]. A common extension of these two classes is the class of graphs that do not contain an odd-K_4 as a subgraph. Here *odd-K_4* means a subdivision of K_4, the complete graph on four nodes, in which all triangles have become odd circuits (cf. Figure 1a). Graphs containing no odd-K_4 are t-perfect [9]. However, there are odd-K_4's that are t-perfect, namely, the good-K_4's: a *good-K_4* is a subdivision of K_4, in which two nonadjacent edges are not subdivided and the other four edges have become even paths (cf. Figure 1b). An odd-K_4 that is not good is called a *bad-K_4*; bad-K_4's are not t-perfect (Lemma 11). The main result of this paper is the following theorem.

THEOREM 1. *If G contains no bad-K_4 as a subgraph, then it is t-perfect.*

We prove this in section 3. One of the main tools is the following decomposition result.

THEOREM 2. *If G is* weakly 3-connected*, i.e., a subdivision of a 3-node-connected simple graph, then it contains no bad-K_4 if and only if one of the following holds:*

- *G contains no odd-K_4;*
- *G is an odd-P_9;*

*Received by the editors June 12, 1996; accepted for publication (in revised form) November 14, 1997; published electronically September 1, 1998. This research was partially supported by project HCM-DONET ERBCHRXCT930090 of the European Community.

http://www.siam.org/journals/sidma/11-4/30636.html

†CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands (bgerards@cwi.nl).

‡Centre for Discrete and Applicable Mathematics, London School of Economics, London WC2A 2AE, UK (bshep@lse.ac.uk).

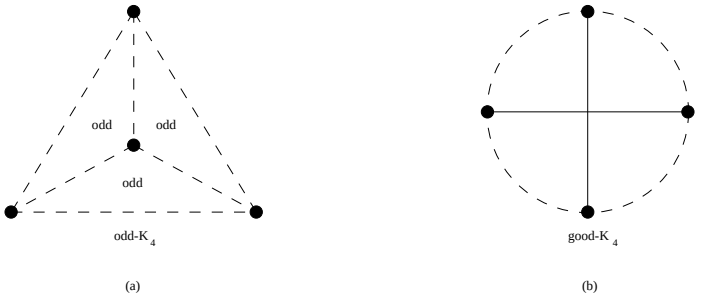


FIG. 1. *Dashed curves indicate internally node disjoint paths of positive length, which in* (b) *all have even length.*

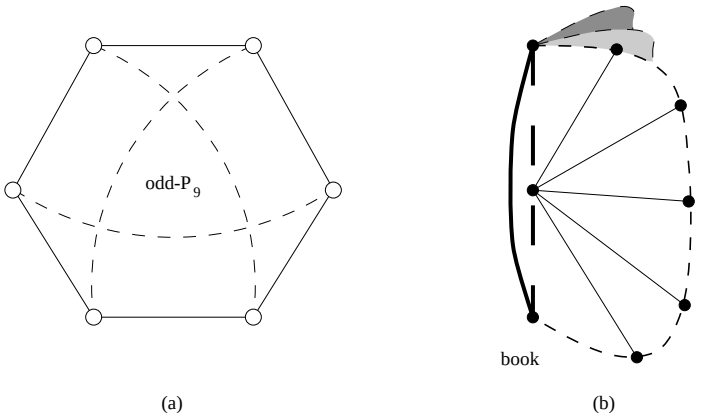


FIG. 2. *Dashed curves indicate internally node disjoint paths of positive even length. The shaded regions in* (b) *indicate the second and third leaf of the book.*

- *G is a clean pad;*
- *G is a book.*

An *odd-P_9* is a graph obtained from a six circuit $u_1u_2, \ldots, u_5u_6, u_6u_1$ by adding three node disjoint even u_iu_{i+3}-paths ($i = 1, 2, 3$); see Figure 2a. Note that the smallest odd-P_9 is the Petersen graph with a node removed.

A *pad* is a graph G with a Hamiltonian circuit $w_1, u_1, w_2, u_2, \ldots, w_k, u_k$ such that an edge not on the Hamiltonian circuit has both end nodes in $U(G) := \{u_1, u_2, \ldots, u_k\}$. (We also define $W(G) := \{w_1, w_2, \ldots, w_k\}$.)

Clearly, a pad has exactly one Hamiltonian circuit, which we denote by $R(G)$ and call the *rim* of the pad. The set of edges not on the rim, called *chords*, will be denoted by $K(G)$. A pad G is *clean* if neither of the two pads in Figure 3 can be derived from G by deleting chords and contracting edges on the rim.

A *book* is any graph that can be constructed as follows:

- Take two nodes h_1 and h_2 (the *hinges* of the book), and join them by an edge.
- Take a third node c, the *center* of the book, and add two internally node disjoint even paths, one from c to h_1 and one from c to h_2 (together with h_1h_2 these paths form the *spine* of the book).
- Add n internally node disjoint even h_1h_2-paths $P_1, \ldots, P_n$, and select on each P_i a nonempty collection T_i of nodes that are an even distance from h_1 on P_i.

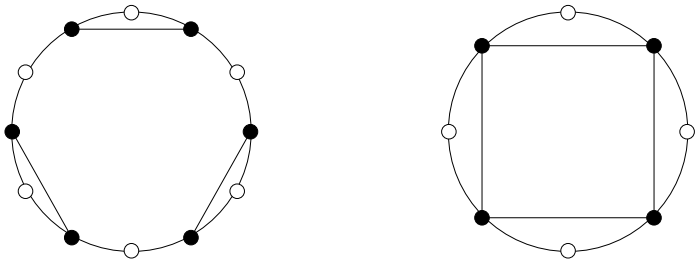


FIG. 3.

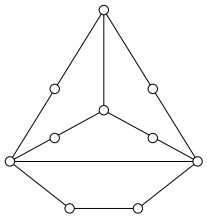


FIG. 4.

- Finally, add all edges in $R_i := \{cr \mid r \in T_i\}$, $i = 1, \ldots, n$.

Note that the union of each $P_i \cup R_i$ with the spine forms a pad. We call these pads the *leaves* of the book. The path P_i is called the *trim* of the leaf. Figure 2b indicates a book with 3 leaves.

As side-product we obtain the following result (we shall give the easy proof in section 2.3).

THEOREM 3. *There exists a polynomial time algorithm that decides whether or not a given graph G contains a bad-K_4.*

Another easy side-product, of which we skip the proof, is that graphs with no bad-K_4 are 3-colorable. This generalizes a result of Catlin [3] that graphs with no odd-K_4 are 3-colorable. Toft [12] conjectures that a graph is 3-colorable if it does not contain a subgraph isomorphic to a graph obtained from K_4 by replacing all six edges with odd paths.

Characterizations around t-perfection. Shepherd [11] characterized which near-bipartite graphs are t-perfect. (A graph is *near-bipartite* if for each node v and each odd circuit C there is a neighbor of v on C. In fact, Shepherd [11] characterized the stable set polytopes of all near-bipartite graphs.) However, the characterization of t-perfection among all graphs is still open.

The graph in Figure 4 is t-perfect—as is easily proved—but contains a bad-K_4, which is not t-perfect. Thus t-perfection is not closed under taking subgraphs. T-perfection is however closed under taking induced subgraphs, i.e., under the deletion of nodes, but a complete list of minimally induced non–t-perfect graphs is not yet known.

However, combining Theorem 1 and Lemma 11, we do have the following:

(2) A graph contains no bad-K_4 if and only if all its subgraphs are t-perfect.

The result of Gerards and Schrijver shows that graphs with no odd-K_4 are t-perfect. In fact, there it is proved that a graph $G = (V, E)$ has no odd-K_4 if and only if for all $a, b \in \mathbb{Z}^V$ and all $c, d \in \mathbb{Z}^E$ the polyhedron

$$\{x \in \mathbb{R}^V \mid a_v \le x_v \le b_v \ (v \in V); c_{uv} \le x_u + x_v \le d_{uv} \ (uv \in E)\} \tag{3}$$

has Chvátal-rank 1, which means that the convex hull of the integral vectors in that polyhedron is obtained by adding all rank-1 Chvátal–Gomory cuts. From Theorem 1 it is not hard to see that a similar result holds for graphs with no bad-K_4.

COROLLARY 4. *$G = (V, E)$ contains no bad-K_4 if and only if for all $a, b \in \mathbb{Z}^V$ and all $c \in \mathbb{Z}^E$ the polyhedron*

$$\{x \in \mathbb{R}^V \mid a_v \le x_v \le b_v \ (v \in V); x_u + x_v \le c_{uv} \ (uv \in E)\} \tag{4}$$

has Chvátal-rank 1.

The rank-1 Chvátal–Gomory cuts needed here are

$$\sum_{v \in V(C)} x_v \le \frac{1}{2} \left\lfloor \sum_{uv \in E(C)} c_{uv} \right\rfloor \ (C \text{ is an odd circuit in } G). \tag{5}$$

One of the main open questions about t-perfection is whether the system of linear inequalities given in (1) is totally dual integral. This property holds for graphs with no odd-K_4 [6], but we have not yet been able to verify this for graphs with no bad-K_4. By the decomposition results used in Gerards [6], it follows that to check for which graphs the system in (1) is totally dual integral for all subgraphs, we may confine ourselves to clean pads and books.

Preliminaries. If G is a graph and u and v are nodes in G of degree at least 3, then a *uv-leg* of G is a uv-path P in G such that all nodes of P, except u and v, have degree 2 in G.

If P is a path in G and $u, v \in V(P)$ we denote the uv-path in P by P_{uv}. If $e = uv \in E(G)$, $P_e := P_{uv}$.

2. Structure of graphs with no bad-K_4. We first prove that if a weakly 3-connected graph with no bad-K_4 contains an odd-K_4, then it is either an odd-P_9, a book, or a pad (Lemma 5). Next we prove that a weakly 3-connected pad with no bad-K_4 is clean (Lemma 6). Together these two lemmas prove the only-if direction of the equivalence in Theorem 2. As odd-P_9's clearly have no bad-K_4, the if direction follows by proving that clean pads (Lemma 7) and books (Lemma 8) have no bad-K_4. We conclude this section with a recognition algorithm for graphs with no bad-K_4.

2.1. Books and pads. Let G be a pad. If H is a subgraph of G and not a pad itself, we denote by $K(H)$ the edges in $K(G)$ with both end nodes in $V(H)$.

If P is a path on $R(G)$, we say that chords e and f are *nested on P*, written as $e \succ_P f$, if $e, f \in K(P)$ and P_f is a subpath of P_e. Chords e, f of $K(G)$ are *nested* if they are nested on some path on $R(G)$; if not, e and f *cross* (notation: $e \times f$).

LEMMA 5. *Let G be a weakly 3-connected graph with no bad-K_4. If G contains an odd-K_4, then G is an odd-P_9, a book or a pad.*

Proof. We first give some definitions: Let H be a subgraph of a graph G. A *route* of H or an *H-route* is a uv-path P in G such that $V(P) \cap V(H) = \{u, v\}$ and such that no leg of H contains both u and v. We say that nodes u_1, u_2, and u_3 *induce an extended triangle* in H if each pair is connected by a leg of H. A collection of three

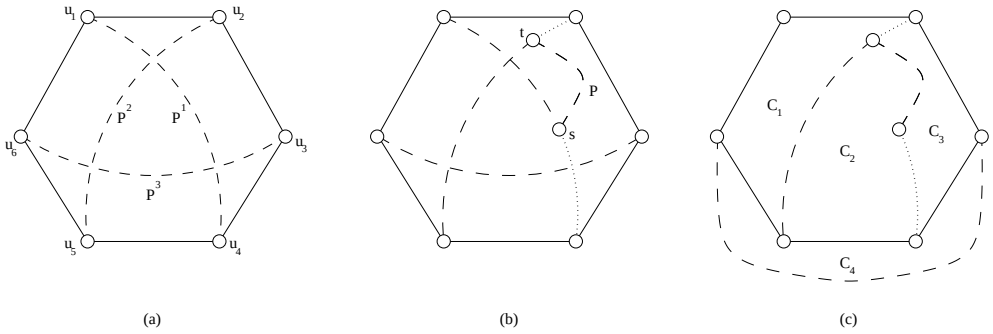


FIG. 5. *Dotted and dashed curves indicate internally node disjoint paths; dashed curves have positive length, whereas dotted curves may have length zero. In* (a), *dashed curves have an even number of edges.*

internally node disjoint vu_i-paths P_i $(i = 1, 2, 3)$ that are internally node disjoint from H is called an *H-tripod* if $v \notin V(H)$ and u_1, u_2, u_3 induce an extended triangle in H.

It is an easy graph theoretical fact that if H is a weakly 3-connected proper subgraph of a weakly 3-connected graph G, then G contains an H-route, or each leg of H is a leg of G and G contains an H-tripod. Moreover, adding an H-route to a weakly 3-connected graph H yields a weakly 3-connected graph.

Assume that G is a counterexample to the lemma with a minimum number of edges.

CLAIM 1. *G contains no odd-P_9.*

Proof of Claim 1. Suppose the claim is false and that H is an odd-P_9 in G. Let $u_1u_2, u_2u_3, \ldots, u_6u_1$ be the six length-1 legs of H; and, for $i = 1, 2, 3$, let P^i be the even u_iu_{i+3}-leg of H (see Figure 5a). By assumption $G \neq H$. As H is weakly 3-connected and has no extended triangle, there exists an H-route P in G. Let s and t be the end nodes of P. One argues that without loss of generality, $s \in P^1 \setminus u_1$ and $t \in P^2 \setminus u_5$ (see Figure 5b). Let $G' := (H \setminus P^1_{u_1s}) \cup P$ and C_1, C_2, C_3, and C_4 be circuits as indicated in Figure 5c. Clearly, C_1 and C_4 are odd circuits. Moreover, C_2 is even, as otherwise the union of C_1, C_2, and C_4 is a bad-K_4. Hence, C_3 is odd, so the union of C_4, C_3, and the symmetric difference of C_1 and C_2 forms a bad-K_4. □

CLAIM 2. *If H is a good-K_4 and P an H-route, then P is an edge and $H \cup P$ is a pad with $R(H \cup P) = R(H)$.*

Proof of Claim 2. H is a pad. Let u_1u_3 and u_2u_4 be the two chords of H and Q^1, Q^2, Q^3 and Q^4 be the four legs of H on $R(H)$ (see Figure 6a). Let s and t be the two end nodes of P. We may assume that $s \in V(Q^1) \setminus \{u_1, u_2\}$ and $t \in V(Q^2) \cup V(Q^3) \setminus \{u_2, u_4\}$. Let C be the unique circuit in $(R(H) \cup P) \setminus Q^4$ (see Figure 6b, c).

First suppose that C is even. If t were in $V(Q^2) \setminus \{u_2\}$ (Figure 6b), then $(H \setminus Q^2_{u_2t}) \cup P$ would be an odd-K_4, with $R_1 := u_1u_3$ and $R_2 := u_4u_2 \cup Q^1_{u_2s}$ as a pair of node disjoint legs. As R_1 has length 1 and R_2 does not, this odd-K_4 would be bad, so $t \in V(Q^3) \setminus \{u_3, u_4\}$ (see Figure 6c). As $H \cup P$ is not an odd-P_9, one of $Q^1_{u_1s}$, $Q^1_{u_2s}$, $Q^3_{u_3t}$, and $Q^3_{u_4t}$ has more than one edge. By symmetry we may assume that this is the case for $Q^1_{u_1s}$. But then all the legs of the odd-K_4 $(H \setminus Q^2) \cup P$, except maybe P or $Q^3_{u_4t}$, have more than one edge. Hence this odd-K_4 is bad.

Therefore, C is odd and thus $H^* := R(H) \cup P \cup \{u_4u_2\}$ is an odd-K_4. Therefore, P has length 1 and H^* is a pad with $R(H^*) = R(H)$. From this it trivially follows

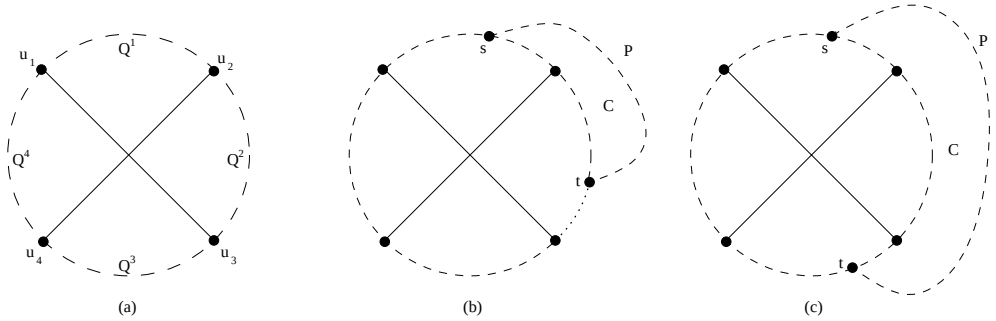


FIG. 6. *Dotted and dashed curves indicate internally node disjoint paths; dashed curves have positive length, whereas dotted curves may have length zero. In* (a), *dashed curves have an even number of edges.*

that also $H \cup P$ is a pad with $R(H \cup P) = R(H)$. □

A pad is called *maximal* if there is no larger pad with the same rim. A subgraph H of G is *induced* if all edges of G with both end nodes in $V(H)$ are in H.

CLAIM 3. *No weakly* 3*-connected maximal pad has a route; hence each one is an induced subgraph of* G *and has a tripod.*

Proof of Claim 3. Let H be a weakly 3-connected pad, and let P be an H-route with end nodes s and t. Let Q_1 and Q_2 be the two st-paths on $R(H)$. As $H' := H \cup P$ is weakly 3-connected, there exists a chord $e = u_1u_2$ of H with $u_i \in V(Q_i) \setminus \{s, t\}$ for $i = 1, 2$. Moreover, as H is weakly 3-connected, there exists a chord f of H crossing e. Now, $H^* := R(H) \cup \{e, f\}$ is a good-K_4. As P is an H-route, $f \neq st$. Thus, s and t lie in different legs of H^*. Hence, by Claim 2, P is an edge—e^*, say—and $H^* \cup e^*$ is a pad with $R(H^* \cup e^*) = R(H^*) = R(H)$. It is trivial to see from this that $H' = H \cup e^*$ is a pad as well. Hence H is not maximal. Therefore, weakly 3-connected maximal pads have no routes.

Now, let H be a weakly 3-connected maximal pad. As it is not equal to G, it must have a tripod. Moreover, if it were not induced, one of its legs would not be a leg of G, but then there would be an H-route. As we have seen, this is not the case, so H is an induced subgraph of G. □

If H is a pad, $u \in V(H)$ is called a *center* of H if the following hold: H has a chord vw such that all other chords cross it and have u as end node, and H has a tripod such that (i) one of its three paths has end node u and this path is of length 1 and (ii) the other two paths end in v and w and are even. We call such a tripod *fitting* H *at* u.

CLAIM 4. *Each weakly* 3*-connected maximal pad has at least one center, and each of its tripods fits at some center of the pad.*

Proof of Claim 4. Let H be a weakly 3-connected maximal pad; let P_1, P_2, and P_3 be the legs of any H-tripod. Denote the end node of P_i on H by u_i. Let Q^{ij} be the u_iu_j-path on $R(H)$ that does not contain the third node in $\{u_1, u_2, u_3\}$. As H is weakly 3-connected, one of Q^{12}, Q^{23}, and Q^{31} is not a leg of H; i.e., one of the legs of the extended triangle induced by u_1, u_2, u_3 is an edge of $K(G)$. Suppose that Q^{13} is not a leg and, consequently, $u_1u_3 \in K(H)$.

(6) If $u_iu_j \in$ K(H), then $P_i \cup P_j$ is an even path.

Indeed, if not then $R(H), P_i \cup P_j$ and one of the chords of H crossing u_iu_j form a

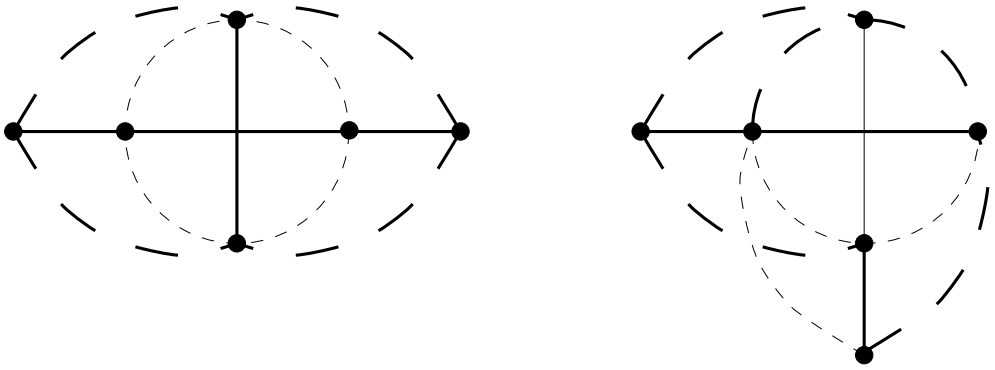


FIG. 7. *Dotted curves indicate internally node disjoint even paths. The bold edges and curves form a bad-K_4.*

bad-K_4.

(7) $P_1 \cup P_2$ and $P_2 \cup P_3$ are odd paths, and so, by (6), Q^{12} and Q^{23} are legs of H.

To see this, let xy be a chord of H crossing u_1u_3; assume $x \in Q^{23}$. It follows from (6) that if (7) were false, then $P_1, P_2, P_3, Q^{23}_{u_2x}, xy, Q^{13}$, and u_1u_3 would constitute a bad-K_4. By (6) and (7), $P_1, P_2, P_3, Q^{12}, Q^{23}$, and u_1u_3 form an odd-K_4, which implies that

(8) P_2 consists of a single edge.

It remains to prove that u_2 is a center of H. Suppose that this is not the case; then there exists a chord e of H with both end nodes in Q^{13} (recall that Q^{12} and Q^{23} are legs of H). But then $P_1, P_2, P_3, Q^{12}, Q^{23}$, and $(Q^{13} \setminus Q^{13}_e) \cup \{e\}$ form a bad-K_4. □

CLAIM 5. *G contains a book with at least two leaves.*

Proof of Claim 5. There exists a weakly 3-connected pad (namely, each good-K_4 is one). As G is not a pad, by Claim 3 there exists a weakly 3-connected pad with no route and hence has a tripod. This pad and that tripod together form a book with two leaves. □

Let $\widetilde{H}$ be a book with center c and hinges v and w, maximum number of leaves $L_1, \ldots, L_n$, and maximum number of edges. Note that for any $i \neq j$, L_i contains an L_j-tripod centered at c. As in the proof of Claim 4, this implies that each chord of L_j has one end in c and the other on the trim of L_j. Moreover, each $V(L_j)$ induces a maximal pad, so by maximality of $\tilde{H}$, each L_j is a maximal pad.

CLAIM 6. *There exists no $\widetilde{H}$-tripod.*

Proof of Claim 6. Let T be a tripod of $\widetilde{H}$. As all extended triangles are contained in leaves, we may assume that T is a tripod of leaf L_1. If T fits L_1 at the center of the book, $\widetilde{H} \cup T$ would be a larger book. Hence T fits L_1 at a node different from c. However, then L_1 has two tripods (namely, T and one in L_2) that fit at different nodes of L_1, so L_1 has at least two centers, which implies that it is a good-K_4. There are two possibilities for how the tripods fit at different nodes (see Figure 7). It is not hard to see that in either case, $L_1 \cup L_2 \cup T$ contains a bad-K_4. □

As G itself is not a book, $\widetilde{H}$ has a route—P, say. Let x and y be the end nodes of P. As the leaves of $\widetilde{H}$ are maximal pads, no one contains both x and y, so we may assume that $x \in V(L_1) \setminus V(L_2)$ and $y \in V(L_2) \setminus V(L_1)$.

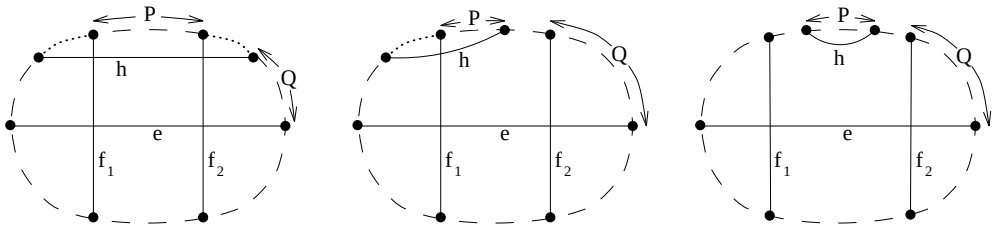


FIG. 8. *Dotted and dashed curves indicate internally node disjoint even paths; dashed curves have positive length, whereas dotted curves may have length zero. The closed curve on the outside is the rim.*

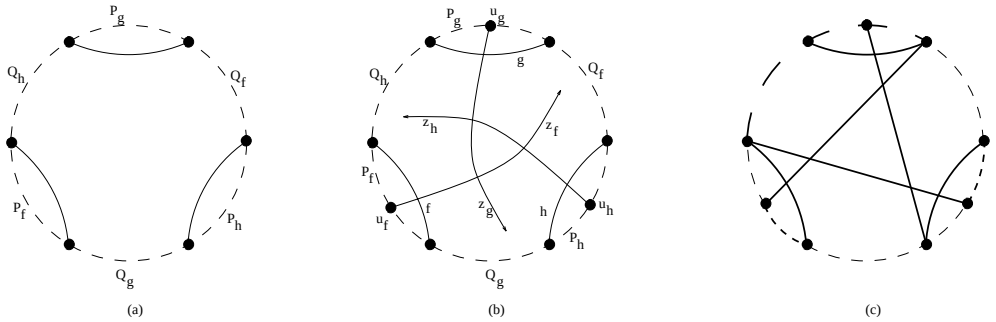


FIG. 9. *Dashed curves indicate internally node disjoint even paths of positive length. The closed curve on the outside is the rim.*

Let Q be the trim of L_2. First, if Q and P do not form a tripod of L_1, then the trim of L_1 contains at least three legs, so L_1 has a route, contradicting Claim 3. Thus Q and P form an L_1-tripod, which—as Q is even—fits at x (by (7)), so P consists of a single edge and L_1 has exactly one chord other than vw, namely, xc. By symmetry, the only chords of L_2 are vw and yc. However, now, xc, yc, xy, and the three even paths in $L_1 \cup L_2$ from v to x, y, and c form a bad-K_4. This yields a final contradiction. □

2.2. Clean pads. Before we can state and prove the next lemma, we need some further definitions. Let G be a pad. Chords e and f *touch*, written as $e \vee f$, if they share an end node. Chords e and f are *parallel* $(e \| f)$ if they are nested but do not touch.

A *mesh* is a collection of four chords e, f_1, f_2, h with the following properties:

- $e \times f_1$, $e \times f_2$, $f_1 \| f_2$, and $h \| e$;
- h is not a chord of any of the four legs on $R(G)$ of the pad $R(G) \cup \{e, f_1, f_2\}$ that are adjacent with e.

There are several possibilities for four chords to form a mesh. They are listed in Figure 8. If we delete the paths P and Q on $R(G)$ indicated in Figure 8, we obtain a bad-K_4. Hence, a pad with no bad-K_4 contains no mesh.

A 3-*chain* is a triple $e, f, g \in K(G)$ such that $e \succ_P f \succ_P g$ for some path P on $R(G)$. A *dirty triple* is a collection of three pairwise parallel edges that do not form a 3-chain (see Figure 9a). A path P on $R(G)$ is *nesting* if each pair of chords on $K(P)$ is nested. G is *nesting* if, for each pair of nodes $s, t \in V(G)$, one of the two st-paths on $R(G)$ is nesting.

It is straightforward to prove that

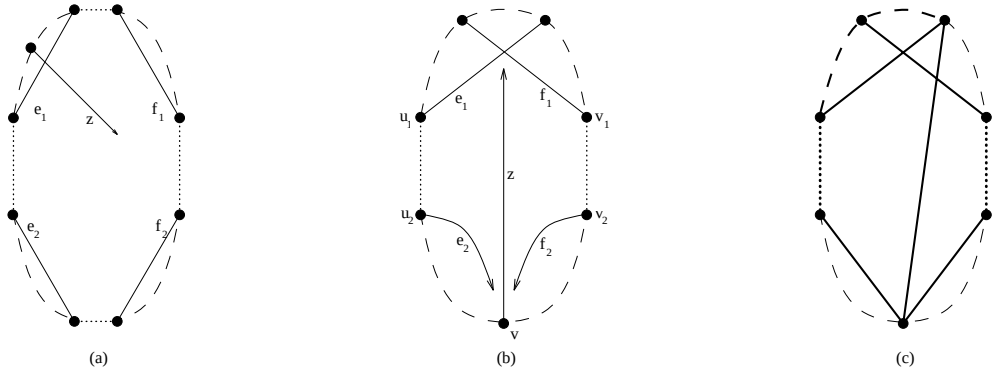


FIG. 10. *Dotted and dashed curves indicate internally node disjoint even paths; dashed curves have positive length, whereas dotted curves may have length zero. The closed curve on the outside is the rim. The bold edges and curves in* (c) *form a test.*

(9) a pad is clean if and only if it is nesting and contains no mesh and no dirty triple.

LEMMA 6. *Each weakly* 3*-connected pad with no bad-*K_4 *is clean.*

Proof. Let G be a weakly 3-connected pad with no bad-K_4. We have already seen that G contains no mesh. Assume that G is not clean.

CLAIM 7. *G is nonnesting.*

Proof of Claim 7. Suppose that G is nesting. Hence, it contains a dirty triple $T := \{f, g, h\}$. Let P_e, Q_e $(e \in T)$ be as in Figure 9a. As G is weakly 3-connected, for each $e \in T$ there exists an edge $z_e := u_e v_e$ crossing e. Assume $u_e \in P_e$ for each $e \in T$. Then, for each $e \in T$, $v_e \in Q_e$, because if v_f, say, were not in Q_f, then z_f, g, f, h would form a mesh or G would be nonnesting.

By symmetry, we may assume that $z_f \| h$. As z_f, z_h, f, h is no mesh, $z_h \vee f$, so $z_h \| g$. Repeating this argument we get that $z_g \vee h$ and $z_f \vee g$. However, now G contains a bad-K_4 (namely, the bold lines in Figure 9c)—a contradiction! □

CLAIM 8. *There exist two edge disjoint paths P_1 and P_2 on $R(G)$ and edges $e_1, f_1 \in K(P_1)$ and $e_2, f_2 \in K(P_2)$ such that*

(i) *e_i and f_i are not nested on P_i $(i = 1, 2)$,*
(ii) *both e_i and f_i share an end node with P_i $(i = 1, 2)$,*
(iii) *$e_1 \times f_1$.*

Proof of Claim 8. By the previous claim, there exist two edge disjoint paths P_1 and P_2 on $R(G)$ and chords e_1, e_2, f_1, and f_2 satisfying (i). It is not hard to see that these paths and chords can be chosen to satisfy (ii) as well. If neither e_1 and f_1 nor e_2 and f_2 are crossing, choose z crossing e_1 (G is weakly 3-connected). With the aid of z, it is straightforward to see that either we can find edge disjoint paths P_1 and P_2 satisfying (i), (ii), and (iii) or we find a mesh (see Figure 9a). As the latter is impossible, the claim follows. □

Choose P_1, P_2, e_1, f_1, e_2, and f_2 as in the previous claim, with $|E(P_1)| + |E(P_2)|$ maximal. Let u_i, v_i be the end nodes of P_i $(i = 1, 2)$. As G is weakly 3-connected, there exists a chord $z = uv$ with $v \in P_2 \setminus \{u_2, v_2\}$ and $u \notin P_2$. By the maximality of $|E(P_1)| + |E(P_2)|$, $u \in P_1 \setminus \{u_1, v_1\}$ (see Figure 10b).

First, consider the special case in Figure 10c. It contains a bad-K_4, indicated by the bold edges. However, the general case, as in Figure 10b, can be transformed to

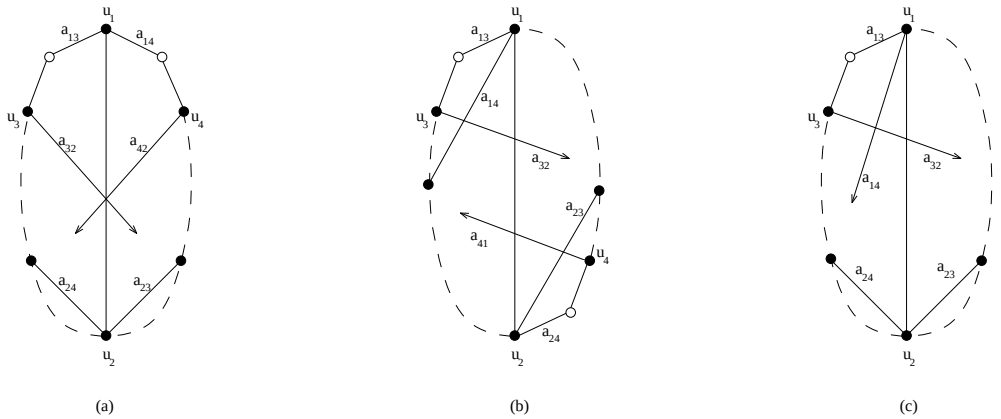


FIG. 11. *Dashed curves indicate internally node disjoint even paths of positive length. The closed curve on the outside is the rim.*

that special case by contracting legs on $R(G)$. As legs are even paths, this contraction could not have created a bad-K_4 if one in G did not already exist. Hence we have a contradiction, so G is clean. □

A chord of a pad is called *universal* if it is not parallel with any other chord.

LEMMA 7. *No clean pad contains a bad-K_4.*

Proof. Let G be a clean pad containing a bad-K_4 H such that $|E(G)|$ is minimal. Let u_1, u_2, u_3, and u_4 be the four nodes of H that have degree three in H. For $i, j = 1, \ldots, 4$, let P^{ij} be the u_iu_j-leg of H.

CLAIM 9. *The following hold:*

(i) $K(G) \subseteq E(H)$.

(ii) *All legs of G on $R(G)$ have length* 2.

(iii) *If P is a leg of H, then $|P \cap R(G)| \leq 2$. If $|P \cap R(G)| = 2$, then P is a leg of G on $R(G)$ or P has length* 3. *In the latter case, the four legs of H meeting P are even, and the sixth leg consists of a single edge.*

(iv) *If $u, v \in U(G)$ form a* 2*-node cutset of G, then there exists a uv-path P on $R(G)$ with* 2 *or* 4 *edges. If P has* 4 *edges, it has one chord, which meets exactly one of u and v.*

Proof of Claim 9. If (i) were false, deleting an edge from $K(G) \setminus E(H)$ would contradict the minimality of G, as would contracting legs of G into legs of length 2 if (ii) were false.

To prove (iii), suppose P is a leg of H that contains edges of $R(G)$. Let e_1 and e_2 be consecutive edges on $P \cap R(G)$. By the minimality of G, contracting e_1 and e_2 in H does not yield a bad-K_4. This means that the leg P of H containing e_1 and e_2, has length 2 or 3. Moreover, in the latter case the four legs of H meeting P are even and the sixth one has length one. Hence (iii) follows.

To see (iv), note that if G has a two node cutset, then H lies mainly on one "side" of that cutset in the sense that one side of the cutset contains at least five legs of H and the other side contains at most (part of) the sixth leg. □

CLAIM 10. *G has no universal chord.*

Proof of Claim 10. Let uv be a universal chord. This means that $G \setminus \{u, v\}$ is bipartite, so uv is a leg of H. Assume $u = u_1$ and $v = u_2$. Let Q^1 and Q^2 be the two uv-paths in $R(G)$. We call $Q^1 \cup K(Q^2)$ and $Q^2 \cup K(Q^1)$ the two *sides* of G. For

$i, j = 1, \ldots, 4$, let a_{ij} be the first edge on P^{ij} going from u_i to u_j. (Thus, $a_{ij} = a_{ji}$ if and only if $|P^{ij}| = 1$.)

As $|P^{12}| = 1$, it follows by Claim 9 that for $i = 1, 2$ and $j = 3, 4$, $a_{ij} \in R(G)$ if and only if P^{ij} is a leg of G in $R(G)$. Moreover, as the circuit $P^{1i} \cup P^{i2} \cup \{u_1u_2\}$ is odd for $i = 3, 4$, we have the following:

(10) If $i = 3, 4$, then a_{1i} and a_{2i} lie on the same side of G. Moreover, P^{1i} and P^{2i} are both even or both odd.

Also, as the circuit $P^{i3} \cup P^{34} \cup P^{4i}$ is odd for $i = 1, 2$, we have that

(11) if $i = 1, 2$, then a_{i3} and a_{i4} lie on different sides of G.

Next we rule out the different cases one by one:

(12) At least one of a_{13}, a_{14}, a_{23}, and a_{24} is in $R(G)$.

Suppose that this is not the case and that $a_{13} \in K(Q^1)$. Then from (10) and (11) it follows that $a_{23} \in K(Q^1)$ and $a_{14}, a_{24} \in K(Q^2)$. Thus both Q^1 and Q^2 are nonnesting, which is a contradiction.

(13) For $i = 1, 2$, either a_{i3} or a_{i4} is in $K(G)$.

To see this, assume that $a_{13} \in Q^1$ and $a_{14} \in Q^2$ (see Figure 10a). By (10), all legs of H adjacent to u_1u_2 are even. Hence P^{34} is odd, and as H is bad, it has at least three edges. By symmetry, we may assume that $Q^2_{u_2u_4}$ is not internally node disjoint with P^{34}. Hence $P^{24} \neq Q^2_{u_2u_4}$. Therefore, by (10), $a_{24} \in K(Q^1)$ and $a_{42} \times u_1u_2$ (by Claim 9(iii) and since u_1u_2 is universal). However, this implies that $P^{23} \neq Q^1_{u_2u_3}$, so, by (10), $a_{23} \in K(Q^2)$ and $a_{32} \times u_1u_2$. If $a_{32} \times a_{23}$, then a_{32}, a_{23}, a_{42}, and a_{24} form a mesh, so a_{32} and a_{23} do not cross. Similarly, a_{42} and a_{24} do not cross. However, this implies that G is nonnesting, a contradiction. Hence (13) follows.

From the above we may assume that $a_{13} \in Q^1$ and $a_{14} \in K(Q^1)$, so P^{23} cannot be $Q^1_{u_2u_3}$. Hence $a_{23} \in K(Q^2)$ and $a_{32} \times u_1u_2$. First assume that $a_{24} \in Q^2$ and consequently $a_{41} \times u_1u_2$ (see Figure 10b). As G is nesting, by symmetry we may assume that $a_{32} \times a_{23}$, but this implies that a_{32}, a_{41}, a_{23}, and a_{14} form a mesh. As G is clean, this is a contradiction, so $a_{24} \notin Q^2$. Hence, $a_{24} \in K(Q^1)$ (see Figure 10c). As a_{32}, a_{41}, a_{23}, and a_{24} is not a mesh, a_{32} does not cross a_{23}. Similarly, a_{24} does not cross a_{14}. But this means that G is nonnesting—a contradiction! □

A chord is *crossed* if it is crossed by at least one other chord. We call chords e_1 and e_2 *distant* if $e_1 \| e_2$ and, for $i = 1, 2$, the path P_i on $R(G)$ with the same end nodes as e_i but node disjoint from e_{3-i} satisfies $K(P_i) = \{e_i\}$.

CLAIM 11. *Each pair of distant chords contains a noncrossed chord.*

Proof of Claim 11. Let e_1 and e_2 be a pair of distant chords. Suppose that e_1 is crossed by z_1 and e_2 by z_2. For $i = 1, 2$, z_i does not cross e_{3-i}, as otherwise, z_i would be universal, or there would be a mesh, or e_1 and e_2 would not be distant. As G is nesting $z_1 \times z_2$. Let x_1 be the end node of z_1 and x_2 be the end node of z_2 such that there exists an x_1x_2-path on $R(G)$, called Q, that is internally node disjoint with e_1 and e_2. Assume z_1 and z_2 are selected such that Q is as short as possible. As G contains no mesh, either $z_1 \vee e_2$ or $z_2 \vee e_1$; assume the latter is the case (see Figure 11).

For $i = 1, 2$, let y_i be a chord parallel with z_i (z_1 and z_2 are not universal). From the fact that G is clean, that e_1 and e_2 are distant, and that Q is minimal, one is able

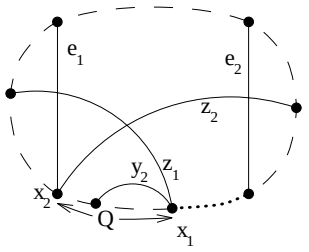


FIG. 12. *Dotted and dashed curves indicate internally node disjoint even paths; dashed curves have positive length, whereas dotted curves may have length zero. The closed curve on the outside is the rim.*

to deduce that $y_2 \in K(Q \setminus x_2)$. As e_1, e_2, y_2 cannot form a dirty triple, y_2 is adjacent to e_2, so x_1 is an end node of e_2. Hence we have symmetry between $i = 1$ and $i = 2$. Therefore, $y_1 \in K(Q \setminus x_1)$ and is adjacent to e_1, but now the edges e_1, e_2, y_1, and y_2 show that G is not nesting—a contradiction! □

If $e = uv$ is a noncrossed chord, then u and v share a common neighbor in $R(G)$ (by Claim 9(iv)), which we denote by u_e. As $e \in E(H)$, the node u_e will not be in $V(H)$.

CLAIM 12. *Each pair of distant chords contains a crossed chord. Moreover, the noncrossed chords in G are pairwise adjacent and there are at most two of them.*

Proof of Claim 12. To prove the first statement, suppose that it is false. Let e and f be two parallel nonadjacent noncrossed chords. Let Q^1 and Q^2 be the two paths on $R(G)$ joining an end of e with an end of f. As G is nesting, we may assume that $K(Q^2) = \emptyset$ and that Q^1 is nesting. As H is contained in $G' = G \setminus \{u_e, u_f\}$, G' is nonbipartite. Hence $K(Q^1) \neq \emptyset$. Let $h \in K(Q^1)$ with Q^1_h minimal. As G has no dirty triple, h is adjacent to e or to f. Thus, let us assume that h and e share an end node—v, say. As Q^1 is nesting all chords in $K(Q^1)$ are in $\delta(v)$. But that means that all odd circuits in G' contain v. This is impossible since not all odd circuits in H can go through a single node.

The second statement easily follows from the first. Indeed, two parallel noncrossed chords are clearly distant by Claim 9(iv), so by the first statement of this claim they cannot exist. Suppose there are three pairwise adjacent noncrossed chords e_1, e_2, and e_3. They cannot meet at a single node, as this would contradict Claim 9(iv), so they form a triangle. Hence $K(G) = \{e_1, e_2, e_3\}$ and $R(G)$ is a circuit of length 6, but that graph has no bad-K_4. □

CLAIM 13. *There is exactly one noncrossed chord.*

Proof of Claim 13. Suppose that this claim is false. Let $e = xy$ and $f = yz$ be two noncrossed chords. Let Q be the xz-path on $R(G)$ not containing y. As e is not universal, $K(Q) \neq \emptyset$. Let $g \in K(Q)$ with Q_g minimal. Let $h \times g$ (by the previous claim, g is crossed). As Q is nesting, $h \in \delta(y)$ and each chord in $K(Q)$ crosses h. Hence, h is universal—a contradiction! □

As there are no universal edges, there exists a pair of distant chords. By Claims 11 and 12 one of the two—$e = uv$, say—is crossed, and the other, f, is not. Let P be the uv-path on $R(G)$ not containing u_f. Let Q^1 and Q^2 be the two paths constituting $R(G) \setminus (P \cup \{u_f\})$. For $i = 1, 2$, let K_i be the collection of edges crossing e with end node in Q^i.

CLAIM 14. *$K(Q^1) = K(Q^2) = \emptyset$, $K_1 \neq \emptyset$, and $K_2 \neq \emptyset$.*

Proof of Claim 14. As G is nesting, (i) $K(Q^1) = \emptyset$ or $K(Q^2) = \emptyset$, (ii) $K(Q^1) = \emptyset$

or $K^2 = \emptyset$, and (iii) $K(Q^2) = \emptyset$ or $K^1 = \emptyset$. From this it is easy to check that if the claim is false, then either $K^1 = \emptyset$ and $K(Q^1) = \emptyset$ or $K^2 = \emptyset$ and $K(Q^2) = \emptyset$. Assume that the latter is the case. Let w be the common end node of P and Q^1. There exists an odd circuit in H not containing w. As $u_f \notin V(H)$, this means that $G \setminus \{u_f, w\}$ is nonbipartite. It is straightforward to check that this implies that $K(Q^1)$ contains a chord parallel with e. Let h be such a chord with Q^1_h minimal. Then e and h are distant, so by Claim 11, h is noncrossed, but this contradicts Claim 13. □

Let Q be the path $R(G) \setminus u_f$. For $i = 1, 2$, let $e_i \in K^i$ with $Q_{e_i} \cap P$ maximal. As e, e_1, e_2, f do not from a mesh, $e_1 \times e_2$ or $e_1 \vee e_2$, so there exists a node—w, say—that lies on $Q_{e_1} \cap P$ and on $Q_{e_2} \cap P$. By Claim 14 this means that w lies on Q_g for each chord $g \in K(Q) = K(G)$. Hence $G \setminus \{w, u_f\}$ is bipartite. As H does not contain u_f, this is a final contradiction. □

LEMMA 8. *No book contains a bad-K_4.*

Proof. Suppose that G is a book, and let H be any odd-K_4 in G. Let C be the spine, h_1 and h_2 be the hinges, and c be the center of G. It is easy to see that for every $e \in E(C)$ there is a node $v \in \{h_1, h_2, c\}$ such that each odd circuit in $G \setminus e$ contains v. Hence H must contain C. Consequently, H should be entirely contained in one of the leaves of G. As all leaves are clean pads, H must be a good-K_4. □

2.3. Recognizing graphs with no bad-K_4. In this section we prove Theorem 3, which says that one can check—in polynomial time—whether or not a given graph G contains a bad-K_4.

First of all, note that odd-P_9's, books and clean pads are easily recognized. Second, a polynomial-time recognition algorithm for the containment of an odd-K_4 is given by Gerards et. al. [8] (cf. Gerards [7]). Hence, by Theorem 2, it suffices to prove that we can find for each graph G in polynomial time a polynomial-length list $\mathcal{L}$ of weakly 3-connected graphs smaller than G such that G contains a bad-K_4 if and only if at least one member of $\mathcal{L}$ contains a bad-K_4. The following two easy-to-prove lemmas show that this is indeed the case.

We need some definitions and notations. If G is a graph, then $[G_1, G_2]_{u,v}$ is called a *split* if G_1 and G_2 are subgraphs of G such that $V(G_1) \cap V(G_2) = \{u, v\}$; $E(G_1)$ and $E(G_2)$ partition $E(G)$, $|E(G_1)|, |E(G_2)| \geq 4$; and neither G_1 nor G_2 is an odd circuit. If G_2 is bipartite and contains an odd uv-path, we call the split *odd*. If G_2 is bipartite and contains an even uv-path, we call the split *even*. If both G_1 and G_2 are nonbipartite, we call the split *strong*.

If u and v are two nodes of a graph H and $\ell \in \mathbb{N}$, then $[H]^\ell_{u,v}$ denotes the graph obtained from H by adding a path from u to v with ℓ edges; we abbreviate this as $[H]^{k,\ell}_{u,v} := [[H]^k_{u,v}]^\ell_{u,v}$.

LEMMA 9. *Let $[G_1, G_2]_{u,v}$ be a split of a 2-connected graph G. Then the following hold:*

- *If $[G_1, G_2]_{u,v}$ is odd, then G contains a bad-K_4 if and only if $[G_1]^3_{u,v}$ contains a bad-K_4.*
- *If $[G_1, G_2]_{u,v}$ is even, then G contains a bad-K_4 if and only if $[G_1]^2_{u,v}$ contains a bad-K_4.*
- *If $[G_1, G_2]_{u,v}$ is strong and G has no odd or even split, then G contains a bad-K_4 if and only if at least one of $[G_1]^{2,3}_{u,v}$ and $[G_2]^{2,3}_{u,v}$ contains a bad-K_4.*

It follows from this lemma that given a graph G we can construct a polynomial-sized list $\mathcal{L}'(G)$ of graphs with no splits such that G has a bad-K_4 if and only if at least one member of the list has a bad-K_4. Therefore, we may restrict ourselves to graphs with no split. It is easy to see that a graph with no split can be obtained from

a 3-connected graph H by replacing some edges in H by a path of length 2 or 3 or by a circuit of length 3 or 5. More precisely, a graph G has no split if and only if there exists a 3-connected graph H and five sets P_1, P_2, P_3, C_3, and C_5 partitioning $E(H)$, such that G can be obtained from H as follows: for each edge $uv \in P_2 \cup C_3 \cup C_5$ add a path from u to v with 2 edges; moreover, for each edge $uv \in P_3 \cup C_5$ add a path from u to v with 3 edges; finally, remove all the edges in $P_2 \cup P_3 \cup C_5$. We denote G by $H(P_1, P_2, P_3, C_3, C_5)$. Note that, given G, it is easy to find H and the proper partition of its edge set.

So we see that a graph with no split only fails to be weakly 3-connected because it may have pairs of "parallel" legs. Clearly, from each such pair of legs a bad-K_4 can use at most one leg. So if we would consider the list of graphs obtainable by dropping a leg from each pair of parallel ones, we do not gain or lose bad-K_4's. The nice thing about the graphs on this list is that they are weakly 3-connected; the bad thing is that there may be exponentially many of them. Fortunately there is an easy way out of this; we do not need to create the whole list.

LEMMA 10. *Let $G = H(P_1, P_2, P_3, C_3, C_5)$ be a graph with no split. Then G contains a bad-K_4 if and only if there exists a $T_3 \subseteq C_3$ and a $T_5 \subseteq C_5$ with $|T_3|+|T_5| \leq 6$, such that the graph $H(P_1 \cup T_3, P_2 \cup (C_3 \setminus T_3) \cup (C_5 \setminus T_5), P_3 \cup T_5, \emptyset, \emptyset)$ contains a bad-K_4.*

(In fact, this lemma remains correct if we replace $|T_3|+|T_5| \leq 6$ with $|T_3|+|T_5| \leq 3$.)

3. T-perfection. The main goal of this section is to prove Theorem 1, but we first show that bad-K_4's are not t-perfect. In the remainder of the paper, for a subset $S \subseteq V(G)$, we use χ_S to denote the incidence vector of S in $\mathbb{R}^{V(G)}$.

LEMMA 11. *No bad-K_4 is t-perfect.*

Proof. First, note that K_4 is not t-perfect, as the vector $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ is in $\mathcal{P}(K_4)$, but obviously not the convex combination of characteristic vectors of stable sets in K_4. Next, note that each bad-K_4 can be reduced to K_4 by repeated application of the following operation: take a node u and contract all the edges incident with u. However, this operation preserves t-perfection, which we easily obtain from the following:

(14) Let G be a graph, $u \in V(G)$, and $x \in \mathbb{R}^{V(G)}$ such that $x_v = 1 - x_u$ for each neighbor v of u. Moreover, let $\widetilde{G}$ be obtained from G by contracting all the edges in $\delta(u)$ into a new node $\widetilde{u}$, and let $\widetilde{x} \in \mathbb{R}^{V(\widetilde{G})}$ be defined by $\widetilde{x}_v := x_v$ if $v \in V(\widetilde{G}) \setminus \widetilde{u}$ and $\widetilde{x}_{\widetilde{u}} := 1 - x_u$. Then x is a vertex of $\mathcal{P}(G)$ if and only if $\widetilde{x}$ is a vertex of $\mathcal{P}(\widetilde{G})$.

Hence no bad-K_4 is t-perfect. □

The proof of Theorem 1 uses the following lemma (the graphs $[G_i]^{\ell}_{u,v}$ are defined in section 2.3).

LEMMA 12. *Let G be a graph with induced subgraphs G_1 and G_2 such that $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$.*

(a) *If $V(G_1) \cap V(G_2)$ induces a clique in G, then G is t-perfect if and only if G_1 and G_2 are t-perfect (Chvátal [4]).*

(b) *If G is 2-connected, G_2 is bipartite, and $V(G_1) \cap V(G_2) = \{u, v\}$ with $uv \notin E(G)$, then if u and v are on the same side of the bipartition of G_2, G is t-perfect if and only if $[G_1]^2_{u,v}$ is t-perfect; otherwise, G is t-perfect if and only if $[G_1]^3_{u,v}$ is t-perfect (Sbihi and Uhry [10]).*

(c) *If G is 2-connected, both G_1 and G_2 are nonbipartite, and $V(G_1) \cap V(G_2) = \{u, v\}$ with $uv \notin E(G)$, then G is t-perfect if and only if $[G_1]^2_{u,v}, [G_1]^3_{u,v}, [G_2]^2_{u,v}$, and $[G_2]^3_{u,v}$ are t-perfect (Boulala and Uhry [2], Gerards [6]).*

In fact, the lemma above can be generalized beyond t-perfection: It has been proved by Chvátal [4]—for case (a)—and Barahona and Mahjoub [1]—for cases (b) and (c)—that one can obtain linear descriptions for the stable set polyhedron recursively through decompositions as in Lemma 12.

Proof of Theorem 1. Let $\widetilde{G}$ be a counterexample to the theorem with $|E(\widetilde{G})|$ minimal. By Lemma 12

(15) $\widetilde{G}$ is weakly 3-connected and each of its legs has at most 3 edges.

Let $\widetilde{x}$ be a fractional vertex of $\mathcal{P}(\widetilde{G})$. An edge $uv \in E(\widetilde{G})$ is *tight* if $\widetilde{x}_u + \widetilde{x}_v = 1$; an odd circuit C is *tight* if $\sum_{v \in V(C)} \widetilde{x}_v = \frac{1}{2}(|V(C)| - 1)$. We denote the collection of tight edges by $\mathcal{T}$ and the collection of tight odd circuits by $\mathcal{C}$.

(16) $$0 < \widetilde{x}_v < 1 \text{ for each } v \in V(\widetilde{G}).$$

Indeed, if $\widetilde{x}_u = 0$, then $\widetilde{G} \setminus \{u\}$ would be a smaller counterexample, and if $\widetilde{x}_u = 1$, u has a neighbor v with $\widetilde{x}_v = 0$.

(17) $\widetilde{x}$ is the unique solution of the system

$$\begin{aligned} x_u + x_v &= 1 && (uv \in \mathcal{T}), \\ \sum_{u \in V(C)} x_u &= \frac{1}{2}(|V(C)| - 1) && (C \in \mathcal{C}), \end{aligned}$$

as otherwise $\widetilde{x}$ would not be a vertex of $\mathcal{P}(\widetilde{G})$. For $V_0 \subseteq V(\widetilde{G})$, we define $\mathcal{T}(V_0) := \{uv \in \mathcal{T} | u \in V_0\}$ and $\mathcal{C}(V_0) := \{C \in \mathcal{C} | V(C) \cap V_0 \neq \emptyset\}$.

(18) $$\text{For each } V_0 \subsetneq V(\widetilde{G}): \ |\mathcal{T}(V_0)| + |\mathcal{C}(V_0)| > |V_0|.$$

If this were not true, the restriction of $\widetilde{x}$ to $V(\widetilde{G}) \setminus V_0$ would be a unique solution of the system

$$\begin{aligned} x_u + x_v &= 1 && (uv \in \mathcal{T} \setminus \mathcal{T}(V_0)), \\ \sum_{u \in V(C)} x_u &= \frac{1}{2}(|V(C)| - 1) && (C \in \mathcal{C} \setminus \mathcal{C}(V_0)). \end{aligned}$$

So $\widetilde{G} \setminus V_0$ would be a smaller counterexample to Theorem 1. From (14), it also follows that

(19) $$\delta(v) \not\subseteq \mathcal{T} \text{ for each } v \in V.$$

Claim 15. *If C is an odd circuit, then $E(C) \cap \mathcal{T}$ contains no matching of size $\frac{1}{2}(|V(C)| - 1)$. If C is an even circuit and $E(C) \cap \mathcal{T}$ contains a perfect matching, then $E(C) \subseteq \mathcal{T}$.*

Proof of Claim 15. Let $M \subseteq E(C) \cap \mathcal{T}$ be a matching with at least $\frac{1}{2}(|V(C)| - 1)$ edges. If C is even, then $\frac{1}{2}|V(C)| = \sum_{uv \in M}(\widetilde{x}_u + \widetilde{x}_v) = \sum_{uv \in E(C) \setminus M}(\widetilde{x}_u + \widetilde{x}_v) \leq \frac{1}{2}|V(C)|$; thus, we have equality throughout,which implies that also edges in $E(C) \setminus M$ are in $\mathcal{T}$. If C is odd, then there is exactly one node $u' \in V(C)$ that is incident

with none of the edges in M, so we have $\widetilde{x}_{u'} = \sum_{v\in V(C)} x_v - \sum_{uv\in M}(x_u + x_v) \leq \frac{1}{2}(|V(C)| - 1) - \frac{1}{2}(|V(C)| - 1) = 0$, which contradicts (16). ∎

CLAIM 16. *Let u and v be two nodes on a circuit $C \in \mathcal{C}$ and P be a uv-path that is internally node disjoint from C. If $\mathcal{T} \cap E(P)$ contains a matching M covering each node in $V(P) \setminus \{u, v\}$, then the unique odd circuit in $C \cup P$ using P is tight.*

Proof of Claim 16. Let Q_1 and Q_2 be the two uv-paths in C, and assume that $P \cup Q_1$ is an odd circuit—C', say. Let N be the largest matching in $E(Q_2)$ with $V(N)\cap\{u, v\} = V(M)\cap\{u, v\}$. Then $\sum_{r\in V(C')} \widetilde{x}_r = \sum_{r\in V(C')\setminus V(M)} \widetilde{x}_r + \sum_{rs\in M}(\widetilde{x}_r + \widetilde{x}_s) = \sum_{r\in V(C)\setminus V(N)} \widetilde{x}_r + |M| \geq \sum_{r\in V(C)\setminus V(N)} \widetilde{x}_r + \sum_{rs\in N}(\widetilde{x}_r + \widetilde{x}_s) - |N| + |M| = \sum_{r\in V(C)} \widetilde{x}_r - |N| + |M| = \frac{1}{2}(|V(C)| - 1) - |N| + |M| = \frac{1}{2}(|V(C')| - 1)$. Thus $C' \in \mathcal{C}$. ∎

A circuit C in a graph G is called *separating* if G has subgraphs G_1 and G_2, each properly containing C, such that $V(G) = V(G_1) \cup V(G_2)$, $E(G) = E(G_1) \cup E(G_2)$, $V(C) = V(G_1) \cap V(G_2)$, and $E(C) = E(G_1) \cap E(G_2)$.

CLAIM 17. *No circuit in $\mathcal{C}$ is separating.*

Proof of Claim 17. Let $C \in \mathcal{C}$ be separating, and let G_1 and G_2 be as indicated just above this claim (with $G = \widetilde{G}$). For $i = 1, 2$, let $\widetilde{x}^i$ be the restriction of $\widetilde{x}$ to $V(G_i)$. As both G_1 and G_2 have no bad-$K_4$and fewer edges than $\widetilde{G}$, they are t-perfect. Therefore, there exists a $K \in \mathbb{N}$, stable sets $S^1_1, \ldots, S^1_K$ in G_1, and stable sets $S^2_1, \ldots, S^2_K$ in G_2 (with possible repetitions) such that

$$\widetilde{x}^1 = \frac{1}{K}(\chi_{S^1_1} + \cdots + \chi_{S^1_K}) \text{and} \widetilde{x}^2 = \frac{1}{K}(\chi_{S^2_1} + \cdots + \chi_{S^2_K}). \tag{20}$$

Consequently,

$$|S^i_j \cap V(C)| = \frac{1}{2}(|V(C)| - 1) \text{for} i = 1, 2 \text{and} j = 1, \ldots, K. \tag{21}$$

For $i = 1, 2$ and $uv \in E(C)$, we denote the number of stable sets S^i_j with $u, v \notin S^i_j$ by $\sigma_i(uv)$. As, $\sigma_i(uv) = \sum_{j=1}^K (1-|S^i_j\cap\{u, v\}|) = K-\sum_{j=1}^K \chi^\top_{\{u,v\}}\chi_{S^i_j} = K-K\chi^\top_{\{u,v\}}\widetilde{x}^i = K(1 - \widetilde{x}^i_u - \widetilde{x}^i_v) = K(1 - \widetilde{x}_u - \widetilde{x}_v)$, we have that

$$\sigma_1(uv) = \sigma_2(uv) \text{ for each } uv \in E(C). \tag{22}$$

By (21) and (22), we can renumber the sets $S^2_1, \ldots, S^2_K$, such that

$$\text{for all } j = 1, \ldots, K,\ S^1_j \cap V(C) = S^2_j \cap V(C). \tag{23}$$

Hence, each $S^1_j \cup S^2_j$ is a stable set in $\widetilde{G}$ and

$$\widetilde{x} = \frac{1}{K}(\chi_{S^1_1\cup S^2_1} + \cdots + \chi_{S^1_K\cup S^2_K}), \tag{24}$$

but this contradicts that $\widetilde{x}$ is a fractional vertex of $\mathcal{P}(\widetilde{G})$. ∎

As $\widetilde{G}$ is not t-perfect, it contains an odd-K_4. So, by (15) and Theorem 2, $\widetilde{G}$ is an odd-P_9, a book or a clean pad. We will deal with these cases separately.

CASE 1. *$\widetilde{G}$ is an odd-P_9.*

By (15), $\widetilde{G}$ is in fact the Petersen graph with a node removed; see Figure 13. Let $S_{3,6} = \{u_3, u_6, u_{14}, u_{25}\}$. By (17), there exists an edge $uv \in \mathcal{T}$ with $S_{3,6} \cap \{u, v\} = \emptyset$ or a $C \in \mathcal{C}$ such that $|S_{3,6} \cap V(C)| < (|V(C)| - 1)/2$. It is easy to check that the only

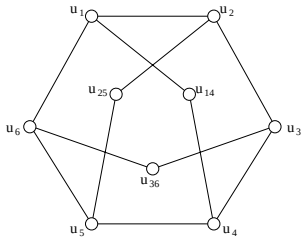


FIG. 13.

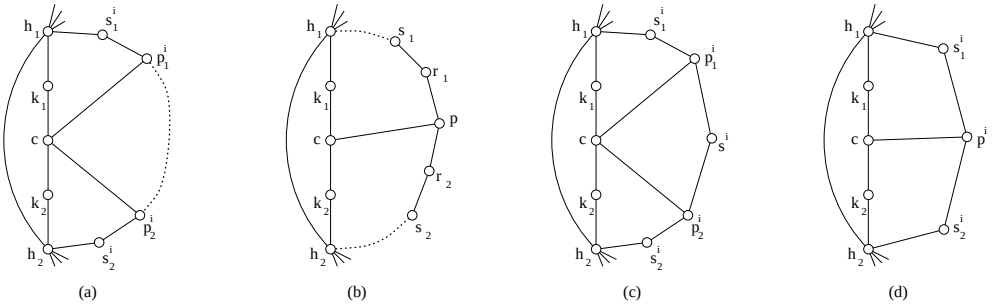


FIG. 14. *Dotted curves indicated internally node disjoint even paths; they may have length zero.*

possibility for this to hold is that either $u_1u_2 \in \mathcal{T}$ or $u_5u_4 \in \mathcal{T}$. By symmetry, we also have $u_2u_3 \in \mathcal{T}$ or $u_6u_5 \in \mathcal{T}$ and $u_3u_4 \in \mathcal{T}$ or $u_6u_1 \in \mathcal{T}$. Again by symmetry, we may assume that $u_1u_2 \in \mathcal{T}$. Hence by Claim 15, $u_6u_5 \notin \mathcal{T}$ and $u_3u_4 \notin \mathcal{T}$. So $u_6u_1 \in \mathcal{T}$ and $u_2u_3 \in \mathcal{T}$. However, that contradicts Claim 15.

CASE 2. *$\widetilde{G}$ is a book.*

Let h_1, h_2 be the hinges of the book and c be the center. Let $L_1, \dots, L_n$ be the trims of the book. By (15), the spine of $\widetilde{G}$ is a circuit of length 5—$h_1k_1ck_2h_2$, say—and the legs of each L_i have length two. Let $h_1s_1^ip_1^i$ be the first leg of L_i and $p_2^is_2^ih_2$ be the last leg of L_i (going from h_1 to h_2; see Figure 14a). It is straightforward to check that

(25) each nonseparating odd circuit in $\widetilde{G}$ is one of $h_1h_2 \cup L_i$; $h_1s_1^ip_1^ick_1$ or $h_2s_2^ip_2^ick_2$ for some $i = 1, \dots, n$.

CLAIM 18. *If $p \in L_i$ and $cp \in E(\widetilde{G})$, then $|\mathcal{C}(p)| \geq 2$. Hence $p \in \{p_1^i, p_2^i\}$.*

Proof of Claim 18. Assume $|\mathcal{C}(p)| \leq 1$. Let s_1r_1p and pr_2s_2 be the two legs of L_i adjacent to p; see Figure 14b. By (18), $|\mathcal{T}(r_1, p, r_2)| \geq 4 - |\mathcal{C}(r_1, p, r_2)| \geq 3$. By (18) and (19), $|\mathcal{T}(r_1)| = |\mathcal{T}(r_2)| = 1$. Hence $cp \in \mathcal{T}$. By (19), we may assume that $pr_1 \notin \mathcal{T}$; hence $r_1s_1 \in \mathcal{T}$. But now the circuit cpr_1s_1 or, if $s_1 = h_1$, the circuit $cpr_1s_1k_1$ violates Claim 15. □

CLAIM 19. *For each $i = 1, \dots, n$, $p_1^i = p_2^i =: p^i$ (see Figure* 14d).

Proof of Claim 19. If not, L_i has three legs; see Figure 14c. By (19), (18), and (25), $|\mathcal{T}(s^i)| = 1$, so, by symmetry, we may assume that $s^ip_2^i \in \mathcal{T}$. By Claim 18, the circuit $ck_1h_1s_1^ip_1^i$ is tight, so by Claim 16, $ck_1h_1s_1^ip_1^is^ip_2^i$ is tight as well. But it has a chord, contradicting Claim 17. □

CLAIM 20. *$ck_1, ck_2 \in \mathcal{T}$.*

Proof of Claim 20. Suppose $ck_2 \notin \mathcal{T}$. Let $S := \{k_1, h_2\} \cup \{p^i | 1 = i, \dots, n\}$.

By (17), there exists an edge $uv \in \mathcal{T}$ with $S \cap \{u,v\} = \emptyset$ or an odd circuit $C \in \mathcal{C}$ with $|S \cap V(C)| < \frac{1}{2}(|V(C)| - 1)$. Using (25), it is easy to check that this implies that $h_1 s_1^i \in \mathcal{T}$ for some $i = 1, \dots, n$. Fix such an i. By (19) and Claim 15, none of $s_1^i p^i, p^i c, p^i s_2^i$, and $s_2^i h_2$ is in $\mathcal{T}$. By (18), $|\mathcal{C}(s_1^i, p^i, s_2^i)| \geq 4 - |\mathcal{T}(s_1^i, p^i, s_2^i)| = 3$, so, $ck_2 h_2 s_2^i p^i \in \mathcal{C}$ and $s_1^i h_1 h_2 s_2^i p^i \in \mathcal{C}$. Hence $\widetilde{x}_c + \widetilde{x}_{k_2} = \widetilde{x}_{s_1^i} + \widetilde{x}_{h_1}$, contradicting that $s_1^i h_1$ is tight and ck_2 is not. □

Now, by (19), we may assume that $cp^1 \notin \mathcal{T}$. By Claims 20 and 15, $\mathcal{T}(s_1^1) = \mathcal{T}(s_1^2) = \emptyset$. Hence $|\mathcal{T}(s_1^1, p^1, s_2^1)| + |\mathcal{C}(s_1^1, p^1, s_2^1)| \leq 3$, contradicting (18).

Case 3. *$\widetilde{G}$ is a clean pad.*

A priori, the tight odd circuits might run quite wildly through $\widetilde{G}$. However, this is not the case, as is shown by the following lemma, which can be understood independently of the present proof.

Lemma 13. *Let C be a nonseparating odd circuit in a clean pad G. Then $|E(C) \cap K(G)| = 1$.*

Proof. Let G be a counterexample with $|E(G)|$ minimal. Let C be a nonseparating odd circuit in G with $|E(C) \cap K(G)| \neq 1$. As contracting all edges on $E(C) \cap R(G)$ yields another counterexample, $E(C) \subseteq K(G)$. Moreover, if $e \in K(G) \setminus E(C)$, then its end nodes lie in different components of $R(G) \setminus V(C)$, as otherwise, $G \setminus \{e\}$ would be a smaller counterexample. We first prove that

(26) $E(C)$ contains no pair of parallel chords.

Indeed, suppose that it is false. Choose parallel chords $f, g \in E(C)$ that are distant in the pad $G \setminus (K(G) \setminus E(C))$. As C is nonseparating, there exist edges $e_f, e_g \in K(G) \setminus E(C)$ with no end node in $V(C)$ such that $e_f \times f$ and $e_g \times g$. If $e_f \| g$ and $e_g \| f$, then G is not clean. Thus, we may assume $e_f \times g$. As C is odd, not all edges on C can cross e_f, so there exists an $h \| e_f$, but then, as f and g are distant in the pad $G \setminus (K(G) \setminus E(C))$, the chords e_f, f, g, and h form a mesh.

Let $c_0, \dots, c_{2k}$ be the nodes of C, numbered in the order in which they lie around $R(G)$. From (26) it then follows that the edges of C are $c_i c_{i+k}$ (indices modulo $2k+1$). Let P_i be the $c_i c_{i+1}$-path on $R(G)$ that contains no nodes of C other than c_i and c_{i+1}, see Figure 14a. Let $K_i := \{uv \in K(G) \setminus E(C) | u \in V(P_i)\}$; note that for each $i = 0, \dots, 2k$, $K_i \neq \emptyset$. For each $e \in K(G) \setminus E(C)$ let C_e be the odd circuit in $R(G) \cup \{e\}$ that uses the fewest nodes of $V(C)$.

(27) If $e, f \in K_i$, then $V(C_e) \cap \{c_i, c_{i+1}\} = V(C_f) \cap \{c_i, c_{i+1}\}$.

Indeed, if not, $c_i c_{i+k+1}, c_{i+1} c_{i+k+1}, e$, and f show that G is nonnesting or has a mesh, and hence is not clean.

From (27), it is easy to see that there exists an $i = 0, \dots, 2k$, such that $V(C_e) \ni c_i$ for all $e \in K_i \cup K_{i-1}$. By circular symmetry, we may assume that $k+1$ is such an i. Let $f \in K_0$. By the symmetry $i \leftrightarrow 2k + 2 - i \pmod{2k+1}$, we may assume that $c_1 \in V(C_f)$; hence $f \| c_0 c_{k+1}$ and $f \times c_1 c_{k+2}$. Let $e \in K_{k+1}$. Then $e \| c_1 c_{k+2}$ and $e \times c_0 c_{k+1}$. Hence $c_0 c_{k+1}, c_1 c_{k+2}, e$, and f form a mesh (see Figure 14b). □

For each $C \in \mathcal{C}$, we denote the unique edge in $E(C) \cap K(\widetilde{G})$ by $k[C]$. Our next task is to study the structure of the collection of tight edges and odd circuits as a whole. The outcome will be summarized in (35), (36), and (37); for proving those we need to derive some claims. We define, for each $\ell = 0, 1, \dots$, $K_\ell := \{e \in K(\widetilde{G}) | e \text{ is in } \ell \text{ tight odd circuits}\}$, $K_\ell^{\text{tight}} := K_\ell \cap \mathcal{T}$, and $K_\ell^{\text{free}} := K_\ell \setminus \mathcal{T}$. By Lemma 13, $K_\ell = \emptyset$ for $\ell \geq 3$. Moreover,

(28) $$K_0^{\text{free}} = \emptyset,$$

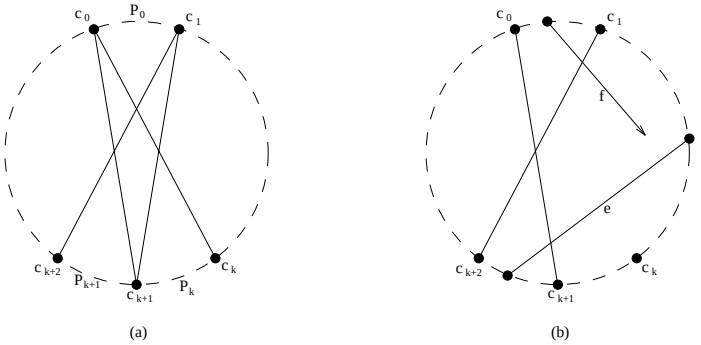


FIG. 15. *Dashed curves indicate internally node disjoint even paths of positive length. The closed curve on the outside is the rim.*

as deleting an edge in K_0^{free} form $\widetilde{G}$ would yield a smaller non–t-perfect graph.

CLAIM 21. *If $uv \in K(\widetilde{G}) \cap \mathcal{T}$, then uv is the only chord with end node u.*

Proof of Claim 21. Let uw be a second chord. Let $P := vv' \dots w'w$ be the vw-path on $R(\widetilde{G})$ not containing u. If there exists a tight odd circuit using both v and w, then by Claim 16, there exists a tight odd circuit using vu and uw, but this contradicts Claim 17 or Lemma 13. Let $C_w \in \mathcal{C}(w')$ and $C_v \in \mathcal{C}(v')$. By Claim 17, $u \notin V(C_w) \cup V(C_v)$, so $k[C_w]$ crosses uw and $k[C_v]$ crosses uv. Hence, $uw, uv, k[C_w]$, and $k[C_v]$ show that $\widetilde{G}$ has a mesh or is nonnesting—a contradiction! □

CLAIM 22. *If $uv \in K(\widetilde{G}) \cap \mathcal{T}$, then uv is not a universal chord of $\widetilde{G}$.*

Proof of Claim 22. Suppose that the claim is false. We construct a new graph G from $\widetilde{G}$ as follows. For each neighbor w of u, we introduce a new node w^* and two new edges uw^* and w^*w and remove the original edge uw. Moreover, we define $x \in \mathbb{R}^{V(G)}$ by $x_w := \widetilde{x}_w$ if $w \in V(\widetilde{G}) \setminus \{u\}$, $x_{w^*} := \widetilde{x}_u$ if w is a neighbor of u in $\widetilde{G}$, and $x_u := 1 - \widetilde{x}_u$. Then, by (14), x is a vertex of $P(G)$.

Let G' be obtained from G by contracting uv^* and v^*v into one new node, called v again. As $x_u + x_{v^*} = 1 = x_{v^*} + x_v$, we get from (14) that G' is not t-perfect. On the other hand, as uv is universal in $\widetilde{G}$, each odd circuit in G' goes through v. However, Fonlupt and Uhry [5] have proved that graphs containing a node that lies on each odd circuit are t-perfect—a contradiction. □

As tight odd circuits have no chords, we have by Claim 21 and (28) that

$$|\delta(u) \cap K(\widetilde{G})| \leq 2 \text{ for all } u \in V(\widetilde{G}) \tag{29}$$

and

$$\text{if } e \in K_2, \text{ then all other chords cross } e. \tag{30}$$

By (30) and Claim 22,

$$K_2^{\text{tight}} = \emptyset. \tag{31}$$

For each $e \in K(\widetilde{G})$ define y_e to be the total number of tight odd circuits and edges containing e. From Claims 21 and 22 and by (29) and (30), we see that

$$\sum_{e \in \delta(u) \cap K(\widetilde{G})} y_e \leq 2 \quad \text{for each } u \in U(\widetilde{G}). \tag{32}$$

Moreover, by (19),

$$|\mathcal{T}(u)| \le 1 \text{ for each } u \in W(\widetilde{G}), \tag{33}$$

and thus, by (17),

$$\begin{aligned} |V(\widetilde{G})| &= |U(\widetilde{G})| + |W(\widetilde{G})| \\ &\ge \frac{1}{2} \sum_{u \in U(\widetilde{G})} \sum_{e \in \delta(u) \cap K(\widetilde{G})} y_e + \sum_{u \in W(\widetilde{G})} |\mathcal{T}(u)| \\ &= \sum_{e \in K(\widetilde{G})} y_e + \sum_{u \in W(\widetilde{G})} |\mathcal{T}(u)| \\ &= |\mathcal{C}| + |\mathcal{T}| \\ &\ge |V(\widetilde{G})|. \end{aligned} \tag{34}$$

Thus, we have equality throughout, which implies that we have equality in (32) and (33). So we get

$$|\mathcal{T}(u)| = 1 \text{ for each } u \in W(\widetilde{G}); \tag{35}$$

(36) each chord in $K_1^{\text{tight}} \cup K_2^{\text{free}}$ is node disjoint from all other chords; moreover, the edges in K_1^{free} form node disjoint circuits;

and, by Claim 22,

$$K_0^{\text{tight}} = \emptyset. \tag{37}$$

As $W(\widetilde{G})$ is a stable set, by (17), there exists an equation in (17) that does not hold for $\chi_{W(\widetilde{G})}$. Case checking yields that this means that

$$K_1^{\text{tight}} \neq \emptyset. \tag{38}$$

Claim 23. $K_1^{\text{free}} \neq \emptyset$.

Proof of Claim 23. Suppose that $K_1^{\text{free}} = \emptyset$. Then, by (36), no two chords touch. By (38), there exists at least one tight chord, so, by Claim 22, there exists a pair of parallel chords. Choose $e, f \in K(\widetilde{G})$ parallel, such that the shortest path—Q, say—on $R(\widetilde{G})$ that connects an end node of e with an end node of f is as short as possible. Let $C_e \in \mathcal{C}(e)$ and $C_f \in \mathcal{C}(f)$ (as e and f are parallel, these two odd circuits are unique). Let $u \in W(G) \cap V(Q)$ and $C \in \mathcal{C}(u)$. (C exists by (18) and (35).) As $u \notin V(C_e) \cup V(C_f)$, $e, f \neq k[C]$, and by the choice of e and f, $k[C]$ is not parallel to e nor to f. Therefore, as there are no touching chords, $k[C]$ crosses both e and f. As $k[C]$ is tight, there exists an edge $h \| k[C]$. As h is not a chord of C_e nor of C_f, $k[C], e, f$, and h form a mesh—a contradiction! □

For each $e \in K_1$ let $C[e]$ be the unique tight odd circuit using e.

Claim 24. K_1^{free} *contains no pair of parallel chords.*

Proof of Claim 24. Let $f_1 = u_1u_2$ and $f_2 = v_1v_2$ be two parallel chords in K_1^{free}; see Figure 15. Let P be the u_2v_1-path on $R(\widetilde{G})$ containing v_2. By symmetry we may assume that P is nesting. Let u_2w be the second edge in K_1^{free} incident with u_2. As

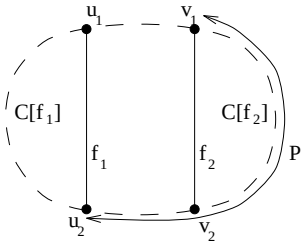


FIG. 16.

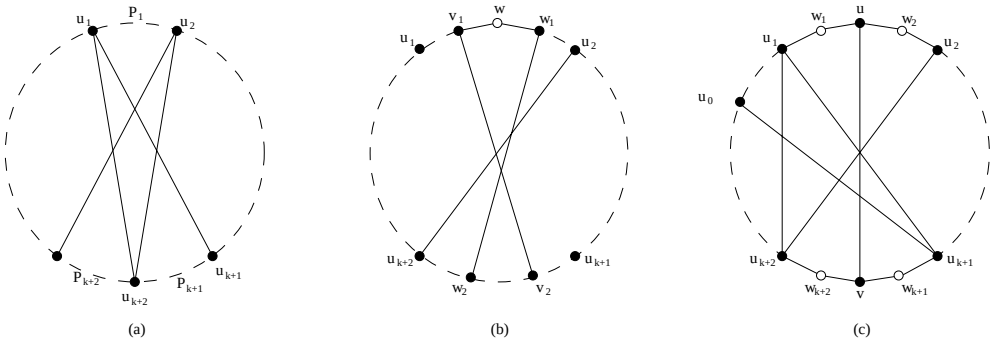


FIG. 17. *Dashed curves indicate internally node disjoint even paths of positive length. The closed curve on the outside is the rim.*

P is nesting, $w \notin P \setminus \{v_1\}$, but then either u_1u_2 or v_1v_2 is a chord of $C[u_2w]$, or u_2w is a chord of $C[u_1u_2]$—a contradiction! □

Let Γ be a circuit in K_1^{free}. Let $u_0, \ldots, u_N$ be the nodes of Γ in the order in which they lie around $R(\widetilde{G})$. From Claim 24, it follows that N is even ($2k$, say) and that the edges in Γ are of the form u_iu_{i+k+1} (indices modulo $2k+1$); see Figure 17a. All chords not in Γ are parallel with at least one edge in Γ. Thus, by (28), (30), and Claim 24, we have that

$$K_2 = \emptyset \text{ and } K_1^{\text{free}} = E(\Gamma). \tag{39}$$

For $i = 0, \ldots, 2k$, let P_i be the u_iu_{i+1}-path on $R(\widetilde{G})$ that is internally node disjoint from Γ. By (38), there exists an edge uv in K_1^{tight}. By symmetry we may assume that $u \in P_1$ and $v \in P_1 \cup \cdots \cup P_{k+1}$. As $C[u_1u_{k+1}]$ has no chords, we have that

$$v \in P_{k+1}. \tag{40}$$

CLAIM 25. *Each chord in K_1^{tight} has one end node in P_1 and one in P_{k+1}.*

Proof of Claim 25. Let $xy \in K_1^{\text{tight}} \setminus \{uv\}$. As we proved for uv, we may assume that $x \in P_i$ and $y \in P_{i+k}$. Hence, $uv \| u_1u_{k+2}$ and $xy \| u_iu_{k+i+1}$. If i were different from 1, then xy, uv,u_1u_{k+2} and u_iu_{k+i+1} would form a mesh or show that $\widetilde{G}$ is nonnesting. Hence $i = 1$ and the claim follows. □

CLAIM 26. $|K_1^{\text{tight}}| = 1$.

Proof of Claim 26. Suppose not; then there are chords v_1v_2 and w_1w_2 in K_1^{tight}, such that u_1 and w_1 are both on P_1 and share a common neighbor w on P_1, see Figure 17b. From (35) we may assume that $v_1w \in \mathcal{T}$, but now the path $v_2v_1ww_1w_2$

and the circuit $C[u_2u_{k+2}]$ satisfy the assumptions in Claim 16. Hence there exists a tight odd circuit using both v_1v_2 and w_1w_2, contradicting Claim 17 or Lemma 13. □

Hence P_1 and P_{k+1} are paths of length 4. Let $P_1 = u_1w_1uw_2u_2$ and $P_2 = u_{k+1}w_{k+1}vw_{k+2}u_{k+2}$; see Figure 17c.

We have that $w_2u_2 \notin \mathcal{T}$, as otherwise the path $vuw_2u_2u_{k+2}$ and the circuit $C[u_0u_{k+1}]$ would satisfy the assumptions of Claim 16 and thus yield a tight odd circuit using three chords of $\widetilde{G}$. By symmetry also $u_1w_1 \notin \mathcal{T}$. Hence, by (35), $uw_1, uw_2 \in \mathcal{T}$. But as $uv \in \mathcal{T}$ this contradicts (19). This completes the proof of Case 3 and thus of Theorem 1. □

Acknowledgments. This research was initiated at the Fourth Bellairs Workshop on Combinatorial Optimization in 1993. The authors sincerely thank the referees for their valuable comments.

REFERENCES

[1] F. Barahona and A. R. Mahjoub, *Compositions of graphs and polyhedra* II: *Stable sets*, SIAM J. Discrete Math., 7 (1994), pp. 359–371.
[2] M. Boulala and J. P. Uhry, *Polytope des indépendants d'un graphe série-parallèle*, Discrete Math., 27 (1979), pp. 225–243.
[3] P. A. Catlin, *Hajós graph coloring conjecture: Variations and counterexamples*, J. Combin. Theory Ser. B, 26 (1979), pp. 268–274.
[4] V. Chvátal, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math., 4 (1975), pp. 305–337.
[5] J. Fonlupt and J. P. Uhry, *Transformations which preserve perfection and h-perfection of graphs*, Ann. Discrete Math., 16 (1982), pp. 83–95.
[6] A. M. H. Gerards, *A min-max relation for stable sets in graphs with no odd-K_4*, J. Combin. Theory Ser. B, 47 (1989), pp. 330–348.
[7] A. M. H. Gerards, *Graphs and Polyhedra—Binary Spaces and Cutting Planes*, CWI Tract 73, CWI, Amsterdam, 1990.
[8] A. M. H. Gerards, L. Lovász, A. Schrijver, P. D. Seymour, C.-H. Shih, and K. Truemper, *Regular matroids from graphs*, in preparation.
[9] A. M. H. Gerards and A. Schrijver, *Matrices with the Edmonds–Johnson property*, Combinatorica, 6 (1986), pp. 365–379.
[10] N. Sbihi and J. P. Uhry, *A class of h-perfect graphs*, Discrete Math., 51 (1984), pp. 191–205.
[11] F. B. Shepherd, *Applying Lehman's theorems to packing problems*, Math. Programming, 71 (1995), pp. 353–367.
[12] B. Toft, *Problem* 10, in Recent Advances in Graph Theory: Proc. Symposium Prague, June 1974, M. Fiedler, ed., Academia, Praha, 1975, pp. 543–544.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 546–556, November 1998

003

MONOCHROMATIC PATHS AND TRIANGULATED GRAPHS*

SHIMON EVEN†, AMI LITMAN†, AND ARNOLD L. ROSENBERG‡

Abstract. This paper considers two properties of graphs, one geometrical and one topological, and shows that they are strongly related. Let G be a graph with four distinguished and distinct vertices, w_1, w_2, b_1, b_2. Consider the two properties, $TRI^+(G)$ and $MONO(G)$, defined as follows.

$TRI^+(G)$: There is a planar drawing of G such that
- all 3-cycles of G are faces;
- all faces of G are triangles except for the single face which is the 4-cycle $(w_1 - b_1 - w_2 - b_2 - w_1)$.

$MONO(G)$: G contains the 4-cycle $(w_1 - b_1 - w_2 - b_2 - w_1)$ and, for any labeling of the vertices of G by the colors {white, black} such that w_1 and w_2 are white, while b_1 and b_2 are black, *precisely one* of the following holds.
- There is a path of white vertices connecting w_1 and w_2.
- There is a path of black vertices connecting b_1 and b_2.

Our main result is that a graph G enjoys property $TRI^+(G)$ if and only if it is minimal with respect to property $MONO$. Building on this, we show that one can decide in polynomial time whether or not a given graph G has property $MONO(G)$.

Key words. planar graphs, triangulated graphs

AMS subject classification. 05

PII. S0895480195283336

1. Introduction.

We consider drawings of simple graphs on the plane and on orientable surfaces. In a drawing $\mathcal{G}$ of a graph G on an orientable surface, a vertex v is represented by a point, and an edge between vertices u and v (denoted $u - v$) is represented by a curve joining its two endpoints. Two such curves do not intersect, except perhaps at their endpoints. When we delete from the surface all points and curves of $\mathcal{G}$, the surface is partitioned to (one or more) connected components called *faces*. If the topological boundary of a face is a cycle of G, we sometimes do not distinguish between the face and the cycle, referring to the cycle as a face.

Let Ψ be a property of graphs. We say that a graph G is *minimal with respect to* Ψ if G satisfies Ψ and any proper subgraph of G does not satisfy Ψ.

A *q-graph* is a simple graph with four distinguished and distinct vertices w_1, w_2, b_1, and b_2 which form a 4-cycle $(w_1 - b_1 - w_2 - b_2 - w_1)$. We refer to this 4-cycle as the *principal cycle* of G and to the edges and vertices of the cycle as the *principal edges* and *vertices*. Other edges and vertices of G are *nonprincipal*.

A *q-path* in a q-graph G is a path[1] whose endpoints are either (w_1, w_2) or (b_1, b_2). A *valid coloring* of G is a labeling of the vertices of G by the colors {white, black} in such a way that vertices w_1 and w_2 are labeled white and vertices b_1 and b_2 are

*Received by the editors March 17, 1995; accepted for publication (in revised form) August 19, 1997; published electronically September 1, 1998.
http://www.siam.org/journals/sidma/11-4/28333.html

†Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel (even@cs.technion.ac.il, litman@cs.technion.ac.il).

‡Department of Computer Science, University of Massachusetts, Amherst, MA (rsnbrg@elys.cs.umass.edu). The research of this author was partially supported by a Lady Davis visiting professorship at the Technion and partially supported by NSF grant CCR-92-21785.

[1] A *path* in a graph G is a sequence of vertices, wherein adjacent vertices are connected by an edge in G. A path is *simple* if no vertex occurs more than once. The *length* of the path is the number of edges, i.e., one less than the number of vertices.

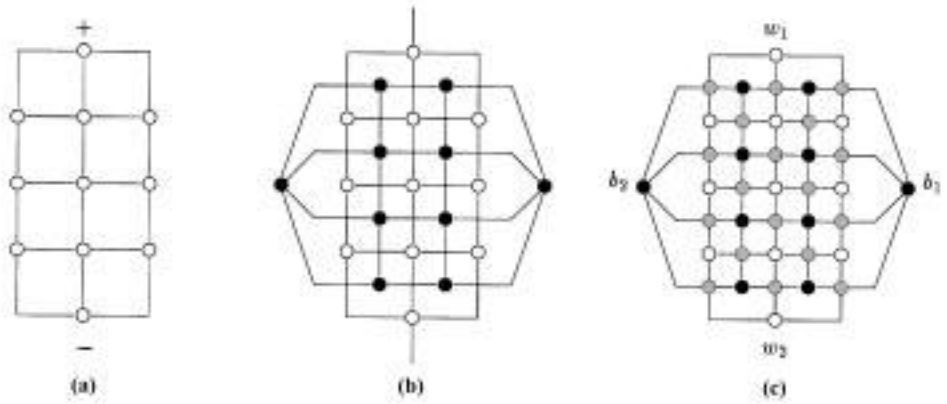


FIG. 1. *Transforming an instance of Shannon's game into a q-game.* (a) *Shannon's game.* (b) *Overlaying the dual graph.* (c) *Replacing crossings with vertices.*

labeled black. This paper is devoted to exposing strong interrelationships among the following properties of q-graphs.

$TRI(G)$: There is a planar drawing of G in which all faces are triangles, except for one face, which is the principal cycle.

$TRI'(G)$: There is a drawing of G on an orientable surface such that all faces are triangles, except for one face which is the principal cycle.

$TRI^+(G)$: There is a planar drawing of G as per property TRI, and, in addition, every 3-cycle is a face. (Thomassen [2] used property TRI^+ to study 2-linked graphs. A graph satisfying TRI^+ is called there a *rib*.)

$MONO'(G)$: Any valid coloring of G has a monochromatic q-path.

$MONO(G)$: $MONO'(G)$ holds and, additionally, no valid coloring of G has both a white q-path and a black q-path.

Our main results demonstrate the following relationships among these properties.

1. *If $TRI'(G)$ holds, then $MONO'(G)$ holds.*
2. *$TRI^+(G)$ holds if and only if G is minimal with respect to property $MONO$.*

Building on these results, we show that one can decide in polynomial time whether or not a given graph G enjoys property $MONO$.

The stimulus for this study comes from a two-player path-construction game that generalizes several other games, namely, Hex, Bridgit, and Shannon Switching Game [1]. This generalized game—let us call it a *q-game*—is played on a q-graph G. The game begins with G in an *initial configuration:*

- Some vertices of G, in particular, b_1 and b_2, are colored black;
- some vertices of G, in particular, w_1 and w_2, are colored white;
- all other vertices of G are uncolored.

The two players, called *Black* and *White*, alternately select an uncolored vertex and color it with their own color. The game concludes when all vertices are colored. Player Black (resp., Player White) wins if there is a black (resp., a white) q-path and no white (resp., no black) q-path in the fully colored G; otherwise, the game is a tie. In this context, property $MONO(G)$ means that there is no tie in a q-game based on the graph G.

Let us see how the q-game generalizes Shannon's Switching Game. Shannon's game is based on a graph K having two distinguished vertices $+$ and $-$. The two

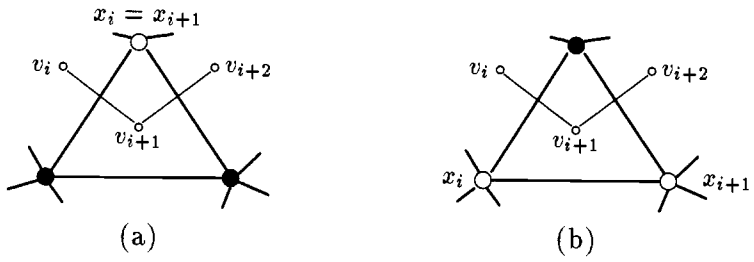


FIG. 2. *Construction of* C_w.

players, called *Short* and *Cut*, alternately select an unclaimed edge and claim it. The game concludes with Short winning if he owns a + to − path; otherwise, Cut wins.

If K is a planar graph, drawn so that vertices + and − lie on the same face, then there is a q-game that generalizes the Shannon game. The q-graph and its initial coloring are constructed as follows. Add to the drawing of K an edge from vertex + to vertex −, drawn within the face that has both vertices. Call the augmented graph K'. On the drawing of K', overlay the (geometric) dual of K'; see Figure 1. Now, replace each edge-crossing in the augmented drawing with a vertex, and add the principal edges (these edges are missing in Figure 1). In the initial configuration, color the vertices of K' white and the vertices of the dual graph black, and leave the crossing vertices uncolored.

2. Property $TRI(G)$ implies property $MONO(G)$.

THEOREM 2.1. *If* $TRI'(G)$ *holds, then* $MONO'(G)$ *holds.*

Proof. Let $\mathcal{G}$ be a drawing of G as provided by property TRI'. Let us augment $\mathcal{G}$ by adding the edge $w_1 - w_2$ drawn within the face $(w_1 - b_1 - w_2 - b_2 - w_1)$. We call the augmented drawing $\mathcal{G}^+$ and, by extension, we call the new depicted graph G^+. (Note that graph G^+ may not be simple.) Clearly, all faces of $\mathcal{G}^+$ are triangles.

Consider now the multigraph G^* that is the geometrical dual [3] of $\mathcal{G}^+$: the vertices of G^* are the faces of $\mathcal{G}^+$, and the edges of G^* are in one-to-one correspondence with edges shared by faces of $\mathcal{G}^+$. For each edge e of G^*, let e° denote the corresponding edge of G^+.

Let a valid coloring of graph G be given. In the resulting colored drawing, an edge or a face of $\mathcal{G}^+$ is called *bichromatic* if it has both black and white vertices. Define the *bichromatic subgraph* $K = \langle V, E\rangle$ of G^* by

$$V = \{v \mid v \text{ is a bichromatic face of } \mathcal{G}^+\},$$
$$E = \{e \mid e^\circ \text{ is a bichromatic edge of } G^+\}.$$

Since every face of $\mathcal{G}^+$ is a triangle, every vertex of K has degree exactly two; hence, K is a collection of disjoint simple cycles.

Let $C = (v_0 - v_1 - \cdots - v_{n-1} - v_0)$ be a cycle of K. Define the "circular list" $C_w \stackrel{\text{def}}{=} (x_0, x_1, \ldots, x_{n-1}, x_0)$ of white vertices of G by

$$x_i = \text{ the white vertex on the edge } (v_i - v_{i+1})^\circ \text{ of } G^+,$$

(where addition on subscripts is modulo n). Let x_i, x_{i+1} be any two consecutive vertices of C_w. If the face v_{i+1} has exactly one white vertex, then $x_i = x_{i+1}$ (see

Figure 2(a)); alternatively, if the face v_{i+1} has two white vertices, then x_i and x_{i+1} are neighbors in G^+ (see Figure 2(b)). Let $\widehat{C}_w$ be the cycle in G^+ obtained by contracting each block of consecutive identical copies of a vertex x in C_w to a single copy of x. ($\widehat{C}_w$ is not necessarily a simple cycle.) Dually, we can define the "circular list" C_b of black vertices of C and the associated contracted cycle $\widehat{C}_b$ of black vertices in G^+.

Consider the triangles $t_1 = (w_1 - w_2 - b_1 - w_1)$ and $t_2 = (w_1 - w_2 - b_2 - w_1)$, which are faces of $\mathcal{G}^+$ and hence are vertices of K. Let C be the cycle of K which contains triangle t_1. On the one hand, if cycle C also contains triangle t_2, then both b_1 and b_2 are in $\widehat{C}_b$ so that $\widehat{C}_b$ contains a path P of black vertices connecting b_1 and b_2. Since path P does not use the edge $w_1 - w_2$, it is a path in the original graph G. On the other hand, if cycle C does not contain triangle t_2, then the edge $w_1 - w_2$ appears exactly once in $\widehat{C}_w$. Hence, $\widehat{C}_w$ contains a path of white vertices connecting w_1 and w_2 which does not use the edge $w_1 - w_2$.

We have thus shown that graph G has property $MONO'(G)$. □

Note that the proof does not use the fact that graph G is drawn on an orientable surface. Hence, Theorem 2.1 holds for graphs drawn on any two-dimensional manifold.

Since a planar q-graph clearly cannot have two disjoint q-paths, one with endpoints (b_1, b_2) and one with endpoints (w_1, w_2), Theorem 2.1 actually implies the following.

LEMMA 2.2. *If $TRI(G)$ holds, then $MONO(G)$ holds.*

3. $TRI^+(G)$ holds if and only if G is $MONO$-minimal. Let G be a graph and Q a set of vertices and edges of G. Let us denote by $G\backslash Q$ the subgraph of G generated by removing all edges of Q, all vertices of Q, as well as their incident edges.

Let G be a q-graph. We say that G' is a *q-subgraph* of G if G' is a subgraph of G and a q-graph (having the same principal vertices as G).

A *trail* T of a q-graph G is a simple path in G such that

1. the endpoints of T are (w_1, w_2) or (b_1, b_2), i.e., T is a q-path.
2. T does not contain the other two principal vertices.
3. for any vertices u and v of T, if u and v are adjacent in G, then u and v are adjacent in T.

LEMMA 3.1. *Let G be a q-graph and P a q-path in G whose only principal vertices are its endpoints. Then there a trail T in G whose vertex-set is a subset of the vertex-set of P.*

Proof. We lose no generality by assuming that P is a b_1-to-b_2 q-path. Let P' be the subgraph of G induced by the vertex-set of P. One verifies easily that any minimal-length b_1-to-b_2 path in P' is a trail in G. □

A consequence of Lemma 3.1 is that any q-graph having property $MONO$ has a trail.

LEMMA 3.2. *Let $MONO(G)$ hold, and let T be a b_1-to-b_2 trail in G. Then*

(a) *w_1 and w_2 are not connected in $G\backslash T$.*

(b) *any simple b_1-to-b_2 path in G whose vertex-set is a subset of T's coincides with T.*

(c) *for any nonprincipal vertex v of T, there is a w_1-to-w_2 trail T' such that v is the only vertex common to T and T'.*

(d) *every q-subgraph of G that has property $MONO(G)$ has trail T as a subgraph.*

Clearly, by symmetry, we may interchange the roles of (w_1, w_2) and (b_1, b_2) in the lemma.

Proof.

(a) If w_1 and w_2 were connected in $G\backslash T$, then one would be able to color G in a way that simultaneously produces both a white and a black q-path, contradicting property $MONO(G)$.

(b) Let $P = (u_1 - u_2 - \cdots - u_m)$ be a simple b_1-to-b_2 path whose vertices all appear in trail $T = (v_1 - v_2 - \cdots - v_n)$. Assume that $P \neq T$, and let i be the smallest index such that $u_i \neq v_i$. Clearly, then, vertices u_{i-1} and u_i are adjacent in G but are not adjacent in T, contradicting the definition of "trail."

(c) Let C be the valid coloring of G whose only black vertices are the vertices of $T\backslash\{v\}$. By (b), G cannot have a black q-path under coloring C. Because $MONO(G)$ holds, then, G must have a white q-path under coloring C; in fact, by Lemma 3.1, G must have a white trail T'. By (a), trail T' must intersect trail T. Since v is the only white vertex of T under coloring C, it must be the only vertex common to trails T and T'.

(d) Assume, for contradiction, that the q-subgraph G' of G has property $MONO(G')$ but does not contain trail T as a subgraph. Let C be the coloring of G' that colors all vertices of $T \cap G'$ black and colors all other vertices white. By (b), G' cannot have a black q-path under coloring C. Since G' has property $MONO(G')$, it must contain a white q-path P under coloring C. However, such a path P would be a path in G that is disjoint from trail T. By (a), such a path cannot exist. ∎

In what follows, we concentrate on b_1-to-b_2 trails for definiteness. The entire development dualizes to w_1-to-w_2 trails by interchanging the roles of the principal sets $\{b_1, b_2\}$ and $\{w_1, w_2\}$.

Let G be a q-graph, and let T be a b_1-to-b_2 trail in G. Define the following subgraphs of G.

$A_1^{(T)} \stackrel{\text{def}}{=}$ the connected component of $G\backslash T$ that contains principal vertex w_1.

$A_2^{(T)} \stackrel{\text{def}}{=}$ the connected component of $G\backslash T$ that contains principal vertex w_2.

$X^{(T)} \stackrel{\text{def}}{=} G\backslash(T \cup A_1^{(T)} \cup A_2^{(T)})$.

Note that the four graphs T, $A_1^{(T)}$, $A_2^{(T)}$, and $X^{(T)}$ form a partition of G: the graphs collectively contain all vertices of G, while property $MONO(G)$ implies that the graphs are disjoint.

Given G and T as above, define the q-graphs $G_1^{(T)}$ and $G_2^{(T)}$ as follows. For $i = 1, 2$, let $G_i' \stackrel{\text{def}}{=} G\backslash(A_i^{(T)} \cup X^{(T)})$. Construct the graph $G_i^{(T)}$ by adding to G_i' the vertex w_i, as well as the edges $w_i - t$ for every vertex t of T. See Figure 3.

LEMMA 3.3. *If $MONO(G)$ holds, and if T is a trail in G, then both $MONO(G_1^{(T)})$ and $MONO(G_2^{(T)})$ hold.*

Proof. By symmetry, it suffices to establish that $MONO(G_1^{(T)})$ holds. To this end, let C be a valid coloring of $G_1^{(T)}$. Extend C to a coloring of G by labeling all vertices of $A_1^{(T)} \cup X^{(T)}$ white.

Now, if $MONO(G)$ holds, then G has a monochromatic q-path P. If P is a black q-path, then P is a subgraph of $G_1^{(T)}$. Alternatively, if P is a white q-path, then by Lemma 3.2a, P intersects T. From the way we have constructed $G_1^{(T)}$, it should be clear that, in this case, we can construct a white q-path in $G_1^{(T)}$ from P. (In short, we replace an initial segment of P by the edge from w_1 to the last vertex of T that appears in P.) We have thus shown that $MONO'(G_1^{(T)})$ holds.

To finish the proof, we must show that $G_1^{(T)}$ never simultaneously has both a black q-path and a white one. Assume for contradiction that, under coloring C, $G_1^{(T)}$

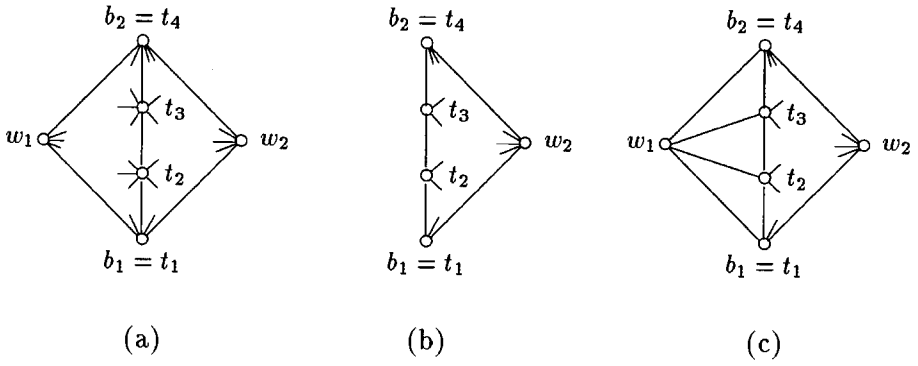


FIG. 3. *Construction of* $G_1^{(T)}$*:* (a) G *and* T, (b) G_1', (c) $G_1^{(T)}$.

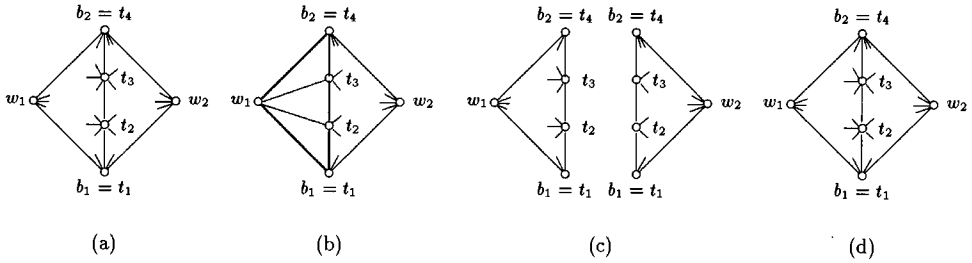


FIG. 4. *Construction of* $\mathcal{G}$*:* (a) G *and* T, (b) $\overline{\mathcal{G}}_1$, (c) $\mathcal{G}_2$ *and* $\mathcal{G}_1$, (d) $\mathcal{G}$.

does simultaneously contain the black q-path $P^{(b)}$ and the white q-path $P^{(w)}$. Then $P^{(b)}$ is a black q-path in G; moreover, Lemma 3.2c assures us that we can use $P^{(w)}$ to construct a white q-path in G that coexists with $P^{(b)}$. This, however, contradicts property $MONO(G)$. □

We are finally ready for our weak converse to Lemma 2.2.

LEMMA 3.4. *If* $MONO(G)$ *holds, then* G *has a q-subgraph* G' *for which* $TRI(G')$ *holds.*

Proof. We prove the lemma by induction on the number of vertices of G. If G has no more than four vertices, then direct inspection verifies that $TRI(G)$ holds. Henceforth, therefore, we assume that G has more than four vertices, and we consider five exhaustive, but not necessarily disjoint, cases.

Case 1. G has a trail T such that both $A_1^{(T)}$ and $A_2^{(T)}$ have at least two vertices.

With no loss of generality, say that T is a b_1-to-b_2 trail. Now, since $MONO(G)$ holds, Lemma 3.3 assures us that both $MONO(G_1^{(T)})$ and $MONO(G_2^{(T)})$ hold also. Let us focus on $G_1^{(T)}$. (A symmetric analysis can be done for $G_2^{(T)}$.) Since $G_1^{(T)}$ has fewer vertices than G, our induction hypothesis guarantees that it has a q-subgraph K_1 that has property $TRI(K_1)$. By Lemma 2.2, then, $MONO(K_1)$ holds. Hence, by Lemma 3.2d, trail T is a subgraph of K_1, and $w_1 - T - w_1$ is a simple cycle in K_1.

Let $\overline{\mathcal{G}}_1$ be a planar drawing of K_1 whose external face is the principal cycle (of G), which goes in the clockwise direction. Generate $\mathcal{G}_1$ from $\overline{\mathcal{G}}_1$ by removing vertex w_1, all edges incident to w_1, and all other vertices and edges that reside in the internal

domain of the plane bounded by the simple cycle[2] $(w_1 - T - w_1)$. See Figure 4. All faces of $\mathcal{G}_1$ are triangles except for the external face $(w_2 - T - w_2)$. In a similar way, construct the planar drawing $\mathcal{G}_2$ whose external face is $(w_1 - T - w_1)$. Merge drawings $\mathcal{G}_1$ and $\mathcal{G}_2$ into a single drawing $\mathcal{G}$ by identifying each vertex of T in $\mathcal{G}_1$ with the corresponding vertex in $\mathcal{G}_2$. Note that since T is a trail, this merging does not duplicate edges of the original graph[3]. Hence, $\mathcal{G}$ is a planar drawing of a q-subgraph G' of G that witnesses property $TRI(G')$.

Case 2. G has a trail of length one.

This case is immediate.

Case 3. G has a trail T of length two.

Say that $T = (b_1 - t - b_2)$ for some vertex t of G. If each $A_1^{(T)}$ and $A_2^{(T)}$ has at least two vertices, then we can just invoke Case 1. Assume, therefore, that one of these graphs, say $A_1^{(T)}$, has only one vertex—which must be principal vertex w_1. In this case, vertex w_1 has precisely three neighbors: vertices b_1, b_2, and t.

Consider the q-graph $G'' \stackrel{\text{def}}{=} G\backslash\{w_1\}$ where t replaces w_1 as a principal vertex. We claim that $MONO(G'')$ holds. To verify this claim, let C be any valid coloring of G''. Extend C to a valid coloring of G by labeling vertex w_1 white. Let P be any monochromatic q-path in G (which must exist by property $MONO(G)$). On the one hand, if P is a black q-path in G, then it is a subgraph of G''; on the other hand, if P is a white q-path in G, then $P\backslash\{w_1\}$ is a white q-path in G''. Now, G'' cannot simultaneously have both a white q-path $P^{(w)}$ and a black q-path $P^{(b)}$, or else G would also have paths of both colors, contradicting property $MONO(G)$. To wit, path $P^{(b)}$ would be a black q-path in G, while path $(w_1 - P^{(w)})$ would be a white q-path in G.

Since G'' has one fewer vertices than G, there is a planar drawing $\mathcal{G}^{(\ni)}$ of a q-subgraph $G^{(3)}$ of G'' that witnesses property $TRI(G^{(3)})$. Let us alter this drawing as follows. In the face $(t - b_1 - w_2 - b_2 - t)$, add vertex w_1 and the edges $w_1 - b_1$, $w_1 - b_2$, and $w_1 - t$. Easily, this altered drawing depicts a q-subgraph $G^{(4)}$ of G that witnesses property $TRI(G^{(4)})$.

Case 4. G has a trail T of length greater than three.

Denote T by $T = (b_1 = t_1 - t_2 - t_3 - \cdots - t_n = b_2)$, where $n > 4$. Lemma 3.2c assures us that there is a w_1-to-w_2 trail T' that shares precisely vertex t_3 with T. It follows that vertex $t_2 \in A_1^{(T')}$ and vertex $t_4 \in A_2^{(T')}$; therefore, Case 1 applies.

Case 5. All trails of G are of length three.

For any vertex v of G, denote by $\Gamma(v)$ the set of neighbors of v, and define

$$s(v) \stackrel{\text{def}}{=} \sum_{w_i \in \Gamma(v)} i + \sum_{b_i \in \Gamma(v)} i.$$

Let C be the valid coloring of G wherein a nonprincipal vertex v is labeled white if and only if $s(v)$ is even. Since G has a monochromatic q-path, it has a monochromatic trail T as well. Let us discuss only the case where T is a b_1-to-b_2 trail; the proof of the other case is similar. Let $T = (b_1 = t_1 - t_2 - t_3 - t_4 = b_2)$. We now infer several important facts about the adjacencies of the vertices of T.

FACT 1. *Vertex t_2 is adjacent to b_1 but not to b_2 and is adjacent to precisely one of w_1, w_2.*

[2] Actually, there are no other vertices and edges, but we do not need this fact.

[3] Otherwise, edges short-circuiting vertices of T may be duplicated.

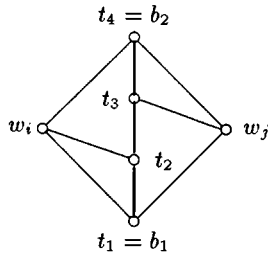


FIG. 5. *The subgraph induced by* $T \cup \{b_1, b_2, w_1, w_2\}$.

By Lemma 3.2c, there is a w_1-to-w_2 trail T' (perforce, of length three) that shares precisely vertex t_2 with trail T. Therefore, vertex t_2 is adjacent to one of w_1 and w_2. Easily, t_2 cannot be adjacent to both w_1 and w_2, nor to both b_1 and b_2, since either such double-adjacency would lead to a trail of length two in G.

FACT 2. *Vertex* t_3 *is adjacent to* b_2 *but not to* b_1 *and is adjacent to precisely one of* w_1, w_2.

This follows by reasoning symmetric to Fact 1.

FACT 3. t_2 *and* t_3 *cannot be adjacent to the same vertex of* $\{w_1, w_2\}$.

Such common adjacency would imply that $s(t_2) \neq s(t_3) \bmod 2$.

Facts 1–3 imply that the subgraph of G induced by the vertices of $T \cup \{b_1, b_2, w_1, w_2\}$ has the form depicted in Figure 5. One can see from this figure that both $A_1^{(T)}$ and $A_2^{(T)}$ must each have at least two vertices. Therefore, the scenario of Case 1 holds.

Since the list of cases above is exhaustive, the lemma is proved. □

We use the next definition and lemma concerning simple graphs rather than q-graphs. For a simple graph K, define the graph $K^{\triangle} = \langle V^{\triangle}, E^{\triangle} \rangle$ by

$$V^{\triangle} \stackrel{\text{def}}{=} \{c |\ c \text{ is a 3-cycle of } K\},$$

$$E^{\triangle} \stackrel{\text{def}}{=} \{c_1 - c_2 |\ c_1 \text{ and } c_2 \text{ share an edge and } c_1 \neq c_2\}.$$

Note that if K' is a subgraph of K, then $(K')^{\triangle}$ is a subgraph of $K^{\triangle}$.

LEMMA 3.5. *Let* K *and* K' *be simple graphs such that* K' *is a subgraph of* K, K' *has at least four vertices, and both* K *and* K' *have planar drawings s.t. every face is a* 3*-cycle and every* 3*-cycle is a face. Then* $K = K'$.

Proof. Consider the graphs $K^{\triangle}$ and $(K')^{\triangle}$. As stated, $(K')^{\triangle}$ is a subgraph of $K^{\triangle}$. Let $\mathcal{K}$ be a planar drawing of K as given by the lemma and let K^* be the geometrical dual of $\mathcal{K}$. Since K has at least four vertices, K^* is isomorphic to $K^{\triangle}$. Hence, $K^{\triangle}$ is connected and is 3-regular.[4] Similarly, the same holds for $(K')^{\triangle}$.

Combining the facts that $(K')^{\triangle}$ is a subgraph of $K^{\triangle}$, both graphs are 3-regular and that $K^{\triangle}$ is connected yields $K^{\triangle} = (K')^{\triangle}$. This implies that K and K' have the same 3-cycles; since each edge of K belongs to some 3-cycle, we must have $K = K'$. □

LEMMA 3.6. *If* $TRI^+(G)$ *holds, then* G *is minimal with respect to property* TRI.

Proof. Assume that $TRI^+(G)$ holds, and let G' be any q-subgraph of G that enjoys property $TRI(G')$. We need to show that $G = G'$. Clearly, G' has a q-subgraph G'' that enjoys property $TRI^+(G'')$. Let $\mathcal{G}$ be a planar drawing of G that

[4] That is, the degree of each vertex is 3.

witnesses property $TRI^+(G)$. By drawing new vertices and edges in the principal face of $\mathcal{G}$, we can construct a planar drawing such that every face is a 3-cycle and every 3-cycle is a face. Let us call the graph depicted by this augmented drawing K. Now, add the same vertices and edges to graph G'', and call the resulting graph K'. K and K' satisfy the requirements of Lemma 3.5; hence, $K = K'$. The equality of K and K', however, implies that $G = G''$. □

The preceding series of lemmas allows us to prove our main theorem.

THEOREM 3.7. *$TRI^+(G)$ holds if and only if G is minimal with respect to property $MONO$.*

Proof. Say first that $TRI^+(G)$ holds. By Lemma 2.2, $MONO(G)$ holds also. Assume then, for contradiction, that G is not minimal with respect to property $MONO$; in particular, let G' be a proper q-subgraph of G that enjoys property $MONO(G')$. By Lemma 3.4, there exists a q-subgraph G'' of G' that enjoys property $TRI(G'')$, but this contradicts Lemma 3.6.

Say next that G is minimal with respect to property $MONO$. Then, by Lemma 3.4, $TRI(G')$ holds for some q-subgraph G' of G. It follows that $TRI^+(G'')$ holds for some q-subgraph G'' of G'. We have just shown, however, that property $TRI^+(G'')$ implies that G'' is minimal with respect to property $MONO$. Since G is also minimal with respect to the property, we must have $G = G''$. □

4. Property *MONO* is decidable in polynomial time. Since property $TRI^+(G)$ can be verified in polynomial time, Theorem 3.7 provides a polynomial-time algorithm for deciding whether or not a given q-graph is minimal with respect to property $MONO$. We need some more technical lemmas to establish that property $MONO$ itself is polynomial-time decidable.

Let G be a q-graph. Define the q-subgraph $\widehat{G} \stackrel{\text{def}}{=} \langle \widehat{V}, \widehat{E} \rangle$ of G, by

$$\widehat{V} \stackrel{\text{def}}{=} \{v \mid v \text{ is a principal vertex or } v \text{ is on some trail}\},$$
$$\widehat{E} \stackrel{\text{def}}{=} \{e \mid e \text{ is a principal edge or } e \text{ is on some trail}\}.$$

The next lemma establishes that any G that enjoys property $MONO$ has exactly one $MONO$-minimal q-subgraph, which is the graph $\widehat{G}$.

LEMMA 4.1. *Assume that $MONO(G)$ holds. Then $\widehat{G}$ is the only q-subgraph of G that is minimal with respect to property $MONO$.*

Proof. By Lemma 3.2d, any q-subgraph of G that enjoys property $MONO$ includes $\widehat{G}$ as a q-subgraph. Hence, we need only establish that $MONO(\widehat{G})$ holds.

To this end, let C be a valid coloring of $\widehat{G}$. Extend C in any way to a valid coloring of G. Since $MONO(G)$ holds, G has a monochromatic q-path; hence, by Lemma 3.1, G has a monochromatic trail. By definition, this trail is a subgraph—hence, a monochromatic q-path—of $\widehat{G}$. Of course, $\widehat{G}$ could not have two conflicting monochromatic q-paths, or else G would also. We conclude that $\widehat{G}$ enjoys property $MONO$. □

Lemma 4.1 implies that, if G is minimal with respect to property $MONO$, then any nonprincipal edge of G is on a trail; moreover, the lemma combines with Lemma 3.2c to imply that any nonprincipal vertex of G is on both a w_1-to-w_2 trail and a b_1-to-b_2 trail.

LEMMA 4.2. *If G is minimal with respect to property $MONO$, then any two distinct nonadjacent vertices of G are separated by a trail.*[5]

[5] Vertices u and v of G are *separated* by trail T if any path connecting u and v contains a vertex of T.

Proof. Let u and v be any two distinct nonadjacent vertices of G. We consider the following two cases.

Case 1. There is a trail T that contains both u and v.

Let x be a vertex of T that lies between u and v. By Lemma 3.2c, there is a trail T' that shares precisely vertex x with T. By Lemma 3.2a, T' separates u and v.

Case 2. No trail of G contains both u and v.

In this case, u and v must be nonprincipal vertices. Let T be a b_1-to-b_2 trail that contains vertex v. Since G is $MONO$-minimal, the arguments of Case 1 in the proof of Lemma 3.4 show that the graph $X^{(T)}$ is empty. Assume, with no loss of generality, that vertex u belongs to the graph $A_2^{(T)}$. Let P be a v-to-w_1 path such that v is the only vertex common to P and T. Consider now the q-graph $G_1^{(T)}$. It is easy to verify by geometrical arguments that $G_1^{(T)}$ enjoys property TRI^+. (Start with a drawing of G that witnesses property TRI^+ and modify it as shown in Figure 3.) Let T' be a b_1-to-b_2 trail in $G_1^{(T)}$ that passes through vertex u, and let P' be a u-to-w_2 path in $G_1^{(T)}$ such that u is the only vertex common to T' and P'.

Let us return now to graph G. T and T' are trails in G which, respectively, avoid vertices u and v; P and P' are paths which both avoid $(T \cup T')\backslash\{u, v\}$. Consider the valid coloring of G wherein vertices of $(T \cup T')\backslash\{u, v\}$ are black and all other vertices are white. Assume that G has a white q-path and, therefore, a white trail T''. Now, trail T'' must intersect both T and T'; hence, it must include both u and v. However, this contradicts the assumption that delineates this case. We conclude, therefore, that G has a black q-path and, therefore, a black trail $\overline{T}$. Since $\overline{T}$ separates vertices w_1 and w_2, and since vertices u and v are connected to w_1 and w_2, respectively, by paths that are disjoint to $\overline{T}$, we see that vertices u and v are separated by $\overline{T}$. □

LEMMA 4.3 (see Thomassen [2]). *If $TRI^+(G)$ holds, then, for any edge e that is* not *in G, the graph $(G \cup \{e\})$ contains two disjoint q-paths.*

Proof. We present here an alternative proof. Let e be the edge $u - v$. By Lemma 4.2, there is a trail T in G that separates vertices u and v. Say, with no loss of generality, that $u \in A_1^{(T)}$ and $v \in A_2^{(T)}$, and that T is a b_1-to-b_2 trail. It follows that G contains a w_1-to-u path P_1 and a v-to-w_2 path P_2, such that both paths avoid trail T. One easily sees that, in the graph $(G \cup \{e\})$, trail T and path $(P_1 - u - v - P_2)$ are disjoint q-paths. □

For any q-graph G, we defined $G^{\oplus}$ to be the graph generated from G by adding a vertex z and four edges connecting z to the principal vertices. For any 3-cycle t in G, let $G^{(t)}$ be the q-subgraph of G generated by removing (from G) all vertices not connected to z in $G^{\oplus}\backslash t$. A q-graph G is *lean* if for any 3-cycle t: $G = G^{(t)}$. (In other words, for any such t, every vertex of $G\backslash t$ is connected (in $G\backslash t$) to some principal vertex.)

LEMMA 4.4. *For any q-graph G and any 3-cycle t in G: property $MONO(G)$ holds if and only if $MONO(G^{(t)})$ holds.*

Proof. Let Y be the set of vertices removed in the construction of $G^{(t)}$. Let P be a q-path in G that uses vertices of Y. Then there is a q-path P' in $G^{(t)}$, whose vertex-set is a subset of P's, that has the same endpoints as P. This means that the vertices of Y are superfluous, as far as monochromatic q-paths are concerned. □

LEMMA 4.5. *Any lean q-graph G enjoys property $MONO(G)$ if and only if it enjoys property $TRI^+(G)$.*

Proof. By Lemma 2.2, property $TRI^+(G)$ implies property $MONO(G)$. To establish the converse, we assume that $MONO(G)$ holds and show that G is $MONO$-

minimal.

Assume, for contradiction, that $G \neq \widehat{G}$. We claim that there is a path P in G whose endpoints are nonadjacent vertices in $\widehat{G}$ and all of whose other ("internal") vertices are not in $\widehat{G}$. If G has an edge that is not in $\widehat{G}$, then this edge is the required path; otherwise, G has a vertex v that is not in $\widehat{G}$. Let X be the connected component of $G \backslash \widehat{G}$ that contains v, and let $\Gamma(X)$ denote the set of neighbors of X in $\widehat{G}$. If $\Gamma(X)$ contains two nonadjacent vertices, then we are done. Alternatively, $\Gamma(X)$ is a clique in $\widehat{G}$. Since $\widehat{G}$ does not include the 4-clique K_4, $\Gamma(X)$ is either empty or is one of the smaller cliques K_1, K_2, or K_3. Since every edge and vertex of $\widehat{G}$ is on a 3-cycle, this contradicts the fact that G is lean. This establishes the existence of the desired path P.

Now, if path P is a single edge, then G contains two disjoint q-paths by Lemma 4.3. However, the same also holds when P is a path of several edges. This contradicts property $MONO(G)$. □

THEOREM 4.6. *Property MONO is polynomial-time decidable.*

Proof. Let a q-graph G be given. By Lemma 4.4, we can reduce G in polynomial time to a lean q-graph G' that has property $MONO$ if and only if G does. Having G', we can decide $MONO(G)$ in polynomial time via Lemma 4.5. □

5. Acknowledgments. We wish to thank Gili Granot for suggesting an alternative proof of Lemma 2.2, and Carsten Thomassen, Tuvi Etzion, and Shmuel Katz for helpful discussions.

REFERENCES

[1] E. R. BERLEKAMP, J. H. CONWAY, AND R. K. GUY, *Winning Ways for Your Mathematical Plays,* Academic Press, New York, 1982.
[2] C. THOMASSEN, 2-*linked graphs*, European J. Combin., 1 (1980), pp. 371–378.
[3] H. WHITNEY, *Planar graphs*, Fund. Math., 55 (1933), pp. 73–84.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 557–589, November 1998

004

TWO ARC-DISJOINT PATHS IN EULERIAN DIGRAPHS*

ANDRÁS FRANK†, TOSHIHIDE IBARAKI‡, AND HIROSHI NAGAMOCHI†

Abstract. Let G be an Eulerian digraph, and let $\{x_1, x_2\}, \{y_1, y_2\}$ be two pairs of vertices in G. A directed path from a vertex s to a vertex t is called an st-path. An instance $(G; \{x_1, x_2\}, \{y_1, y_2\})$ is called feasible if there is a choice of h, i, j, k with $\{h, i\} = \{j, k\} = \{1, 2\}$ such that G has two arc-disjoint $x_h x_i$- and $y_j y_k$-paths. In this paper, we characterize the structure of minimal infeasible instances, based on which an $O(m + n \log n)$ time algorithm is presented to decide whether a given instance is feasible, where n and m are the number of vertices and arcs in the instance, respectively. If the instance is feasible, the corresponding two arc-disjoint paths can be computed in $O(m(m + n \log n))$ time.

Key words. Eulerian digraph, disjoint paths, minimum cut, planar graph, polynomial time algorithm

AMS subject classifications. 05C35, 05C40

PII. S0895480196304970

1. Introduction.

Finding a set of edge-disjoint paths connecting pairs of specified vertices (called terminals) in a graph or a digraph is one of the classical and fundamental problems in graph theory (see [6] for a survey), which has a wide variety of applications. A path between terminals s and t (or a directed path from s to t) is called an st-path. If the graph is undirected, an important result by Robertson and Seymour [10] says that edge-disjoint paths for k pairs $\{s_i, t_i\}$ of terminals, $i = 1, 2, \ldots, k$, can be obtained in polynomial time for a fixed k. In the case of $k = 2$, a complete characterization of undirected graphs G that do not have edge-disjoint s_1t_1- and s_2t_2-paths is available (Dinits and Karzanov [2, 3], Seymour [11], and Thomassen [12]). Such G can be reduced to a graph G' that has a planar representation with the following properties (see Fig. 1):

(i) the four terminals have degree 2, and all other vertices are of degree 3, and

(ii) the terminals are located on the outer face in the order of s_1, s_2, t_1, t_2.

Contrary to this, the characterization of arc-disjoint path problems in digraphs seems much more difficult. For example, the weak 2-linking problem (i.e., to decide whether there are arc-disjoint s_1t_1- and s_2t_2-paths) in a general digraph is shown by Fortune, Hopcroft, and Wyllie [4] to be NP-complete. However, if the digraph under consideration is Eulerian, the situation becomes slightly easier. For a given digraph $G = (V, E)$ with ordered terminal pairs (s_i, t_i), $i = 1, 2, \ldots, k$, call $H = (V, \{(t_i, s_i) \mid i = 1, 2, \ldots, k\})$ its demand digraph. The weak 2-linking problem in an Eulerian digraph $G + H$ is known by Frank [5] to be polynomially solvable. Furthermore, Ibaraki and Poljak [9] showed that the weak 3-linking problem for an Eulerian digraph $G + H$ can also be solved in polynomial time. It is based on the observations that the weak 3-linking problem is equivalent to finding arc-disjoint x_1x_2-, x_2x_3-, and x_3x_1-

*Received by the editors June 10, 1996; accepted for publication (in revised form) June 26, 1997. Published electronically September 1, 1998. This research was partially supported by the Scientific Grant-in-Aid, by the Ministry of Education, Science, Sports, and Culture of Japan, and by the Inamori Foundation.

http://www.siam.org/journals/sidma/11-4/30497.html

†Department of Computer Science, Mathematical Institute, Eötvös University, Múseum körút 6-8, Budapest VIII, Hungary 1088 (frank@cs.elte.hu).

‡Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501 Japan (ibaraki@kuamp.kyoto-u.ac.jp, naga@kuamp.kyoto-u.ac.jp).

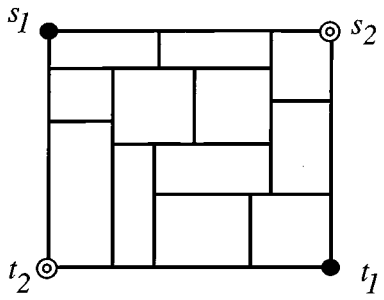


FIG. 1. *An infeasible instance for a two edge-disjoint path problem in an undirected graph G.*

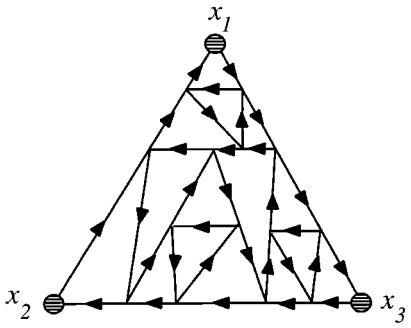


FIG. 2. *An infeasible instance for a weak 3-linking problem in an Eulerian digraph $G+H$.*

paths in an Eulerian digraph with terminals x_1, x_2, x_3 and that the resulting problem is infeasible if and only if it is reducible to a 2-connected Eulerian digraph G', which has a planar representation (see Fig. 2) such that

(i) all terminals have degree 2, and all other vertices have degree 4, and

(ii) every face is a directed cycle, and all the terminals are located on the outer face (which is also a directed cycle) in the order of x_3, x_2, x_1 (where the arcs in the outer face are directed clockwise).

In this paper, we generalize the above result to the two arc-disjoint path problem in an Eulerian digraph G (but $G+H$ is not Eulerian), which decides whether there are arc-disjoint paths connecting two *unordered* terminal pairs $\{x_1, x_2\}$ and $\{y_1, y_2\}$ (i.e., $x'x''$- and $y'y''$- paths, where either $x'x'' = x_1x_2$ or x_2x_1 and either $y'y'' = y_1y_2$ or y_2y_1). This problem includes the above weak 3-linking problem as a special case: for a given instance $(G; (s_1, t_1), (s_2, t_2), (s_3, t_3))$ of the 3-linking problem (where $G+H$ is Eulerian), add four new vertices x_1, x_2, y_1, and y_2 together with seven new arcs $(t_1, y_1), (y_1, s_2), (t_2, y_2), (y_2, x_1), (x_1, s_3), (t_3, x_2), (x_2, s_1)$ to obtain an instance $(G'; \{x_1, x_2\}, \{y_1, y_2\})$ of the two arc-disjoint path problem (where G' is Eulerian), which is clearly feasible if and only if the instance $(G; (s_1, t_1), (s_2, t_2), (s_3, t_3))$ is feasible. We show that the problem can be solved in $O(m + n \log n)$ time, where m and n are, respectively, the numbers of arcs and vertices in G, by deriving an analogue of the above structural characterization of infeasible instances: an Eulerian digraph G with four terminals x_1, x_2, y_1, y_2 is infeasible if and only if it is reducible to an Eulerian digraph G' that has a planar representation (see Fig. 3(a),(b)) such that

(i) all terminals have degree 2, and all other vertices have degree 4,

(ii) there is at most one cut vertex, and

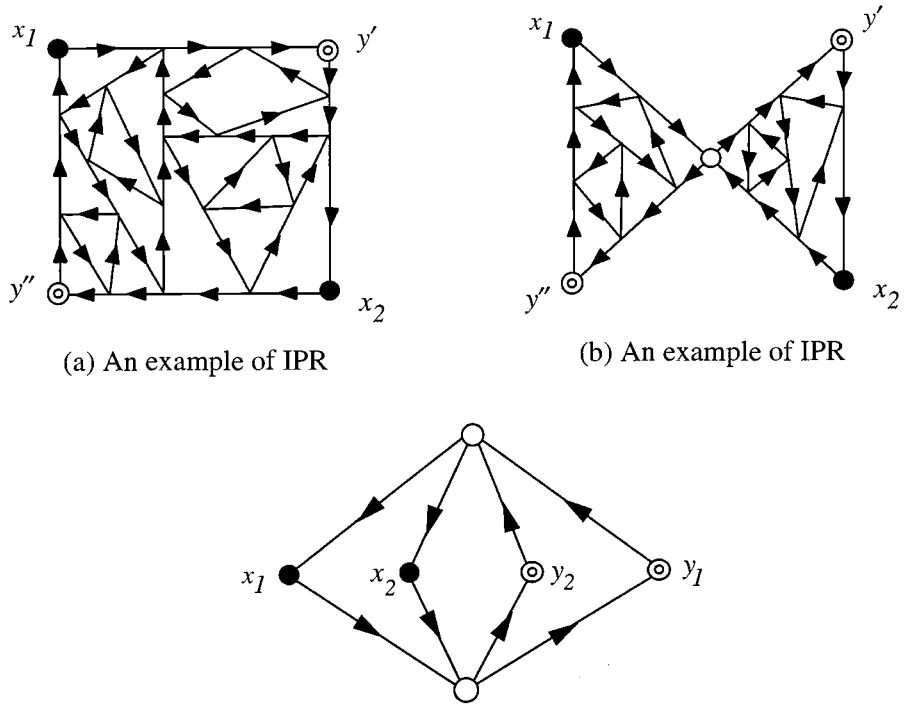


FIG. 3. *Examples of infeasible instances for two arc-disjoint path problems in an Eulerian digraph G.*

(iii) every face is a directed cycle, and all terminals are located on the outer face in the order of x_1, y', x_2, y'', where $\{y', y''\} = \{y_1, y_2\}$.

The proof for this is, however, substantially different from that of [9]. For a feasible instance, we also show that the corresponding two arc-disjoint paths can be computed in $O(m(m + n \log n))$ time.

2. Preliminaries. Let $G = (V, E)$ be a digraph which may have multiple arcs. Denote by $deg(v)$, $indeg(v)$ and $outdeg(v)$ the degree, indegree, and outdegree of a vertex v in G, respectively, where the *degree* of a vertex is the sum of its out- and indegrees. We call a digraph *Eulerian* if the outdegree and indegree of each vertex are equal. Under a *path* or a *cycle*, we always understand a *directed* path or cycle. Repetition of arcs is not allowed, but repetition of vertices is allowed. A cycle that visits every arc exactly once is called Eulerian. A path from s to t is called an st-path. If $\{P_1, P_2, \ldots, P_k\}$ is a collection of arc-disjoint paths such that the last vertex of P_i coincides with the initial vertex of P_{i+1} for each $i = 1, 2, \ldots, k-1$, we denote by $P = \langle P_1, P_2, \cdots, P_k \rangle$ the concatenation of the paths. In the following discussion, digraph G, path P, or cycle C may sometimes be treated either as a vertex set or an arc set, as far as its meaning is unambiguous from the context. If it is necessary to specify, we use $E(G)$ and $V(G)$ to mean the arc set and the vertex set of a digraph G, respectively. For a digraph $G = (V, E)$ and an arc set $E' \subseteq E$, we denote the digraph $(V, E - E')$ by $G - E'$. For a vertex set $Z \subset V$, the subdigraph *induced* by Z is denoted by $G[Z] = (Z, E_Z)$, where $E_Z = \{(u, v) \in E \mid u, v \in Z\}$, and $G[V - Z]$ may be denoted by $G - Z$.

For a subset Z of vertices, $\delta^+(Z)$ denotes the set of arcs from Z to $V-Z$, $\delta^-(Z)$ the set of arcs from $V-Z$ to Z, and $\delta(Z)=\delta^+(Z)\cup\delta^-(Z)$. If G is Eulerian, then $|\delta^+(Z)|=|\delta^-(Z)|$ holds for every Z, where $|A|$ denotes the cardinality of a set A, and therefore $|\delta(Z)|$ is always even. A set $Z\subset V$ is called a *k-cut* if $|\delta(Z)|$ is k. For two disjoint $S,T\subset V$, we say that a cut Z *separates* S and T if $S\subseteq Z$ and $T\subseteq V-Z$ and define $\delta(S,T)$ to be the set of arcs from S to T and arcs from T to S. Throughout this paper, a singleton set $\{v\}$ may also be denoted as v.

Two cuts Z_1 and Z_2 *intersect* each other if $Z_1\cap Z_2\neq\emptyset$, $Z_1-Z_2\neq\emptyset$, and $Z_2-Z_1\neq\emptyset$, and they *cross* each other if, in addition, $V-(Z_1\cup Z_2)\neq\emptyset$ holds. For two crossing cuts Z_1 and Z_2, we easily see that

$$|\delta(Z_1)|+|\delta(Z_2)|\geq|\delta(Z_1\cap Z_2)|+|\delta(Z_1\cup Z_2)| \tag{2.1}$$

and

$$|\delta(Z_1)|+|\delta(Z_2)|=|\delta(Z_1-Z_2)|+|\delta(Z_2-Z_1)|+2|\delta(Z_1\cap Z_2,V-(Z_1\cup Z_2))| \tag{2.2}$$

hold.

Some further notions, such as planarity, edge connectivity, and vertex connectivity, we refer to the unoriented graph $\overline{G}$ obtained from G by ignoring arc orientation. A digraph G is called *connected* if $\overline{G}$ is connected. For a connected digraph $G=(V,E)$, a vertex z is called a *cut vertex* if $G-\{z\}$ has more than one connected component. We call an *undirected* path in $\overline{G}$ a *chain*. A chain with end vertices s and t is called an st-chain, and these s and t are said to be connected (by the chain). A concatenation of a collection of chains is also defined analogously to paths.

Consider an instance $(G=(V,E);X,Y)$ with $X=\{x_1,x_2\}\subseteq V$ and $Y=\{y_1,y_2\}\subseteq V$. Throughout this paper, when we refer to an instance $(G;X,Y)$, we assume that G is Eulerian and is connected (hence strongly connected since G is Eulerian). Each $t\in X\cup Y$ is called a *terminal*. We say that an instance $(G;X,Y)$ is *feasible* if it has two arc-disjoint $x'x''$- and $y'y''$-paths such that $\{x',x''\}=X$ and $\{y',y''\}=Y$; otherwise it is *infeasible*.

LEMMA 2.1. *Let P_X be an $x'x''$-path with $\{x',x''\}=X$ in $(G;X,Y)$. If y_1 and y_2 are connected in $G-E(P_X)$, then $(G;X,Y)$ is feasible.*

Proof. Since G is Eulerian, $G-E(P_X)$ has an $x''x'$-path P'_X and each of the connected components in $G-E(P_X)-E(P'_X)$ is Eulerian (possibly, a single vertex). If y_1 and y_2 are contained in the same connected component in $G-E(P_X)-E(P'_X)$, then the instance is feasible. Assume therefore that y_1 and y_2 are contained in two distinct components H_1 and H_2, respectively. Since y_1 and y_2 are connected in $G-E(P_X)$ but are not connected in $G-E(P_X)-E(P'_X)$, H_1 and H_2 must contain vertices v_1 and v_2 in $V(P'_X)$, respectively. Without loss of generality, assume that P'_X visits v_1 before v_2. Then, H_1 has a y_1v_1-path, P'_X contains a v_1v_2-path, and H_2 has a v_2y_2-path. This implies that the instance is feasible. □

LEMMA 2.2. *If an instance $(G;X,Y)$ satisfies $X\cap Y\neq\emptyset$, then it is feasible.*

Proof. Assume without loss of generality that $x_1=y_1$ for $X=\{x_1,x_2\}$ and $Y=\{y_1,y_2\}$. Since G is connected, there is arc-disjoint x_1x_2-path P_X and x_2x_1-path P'_X. Consider the connected component containing y_2 in $G-E(P_X)-E(P'_X)$. It contains a vertex in $V(P_X)$ or $V(P'_X)$ (say, $V(P'_X)$) since G is connected. Then, y_1 and y_2 are connected in $G-E(P_X)$. Lemma 2.1 then implies that $(G;X,Y)$ is feasible. □

In the following, therefore, we assume $X\cap Y=\emptyset$ for an instance $(G;X,Y)$.

DEFINITION 2.1. *We say that an instance* $(G; X, Y)$ *with* $X = \{x_1, x_2\}$ *and* $Y = \{y_1, y_2\}$ *has an infeasible planar representation* (*IPR*) *if the following conditions hold* (*see Fig.* 3(a),(b)).

(i) *G is planar and has at most one cut vertex.*

(ii) *All the terminals have degree* 2*, and all other vertices have degree* 4.

(iii) *G has a planar representation in which every face is a directed cycle* (*or equivalently, the arcs incident to a vertex are alternately oriented out and in*)*, and all the terminals lie on the boundary of the outer face* (*which is also a directed cycle*) *in the order of* x_1, y', x_2, y''*, where* $Y = \{y', y''\}$. □

We then have the next lemma.

LEMMA 2.3. *Any* $(G; X, Y)$ *which has an IPR is infeasible.*

Proof. If an IPR has arc-disjoint $x'x''$- and $y'y''$-paths, where $\{x', x''\} = X$ and $\{y', y''\} = Y$, then these two paths must cross at some nonterminal vertex in the planar representation (since every terminal has degree 2 and is located on the boundary of the outer face). However, the two paths cannot cross at a nonterminal vertex, because the arcs incident to a vertex are alternately oriented out and in. □

It is easy to see that a feasible instance $(G; X, Y)$ never becomes infeasible by contracting any arc. We say that an instance $(G; X, Y)$ is *minimal infeasible* if it is infeasible, but the instance $(G'; X, Y)$ obtained by contracting *any* arc becomes feasible. The main contribution of this paper is to show that the converse of Lemma 2.3 holds for such minimal infeasible instances. In the case of $|V| = 6$, however, there is a minimal infeasible instance with $V = \{x_1, x_2, y_1, y_2, v, w\}$ and $E = \{(v, x_1), (x_1, w), (v, x_2), (x_2, w), (w, y_1), (y_1, v), (w, y_2), (y_2, v)\}$ (see Fig. 3(c)), which is clearly infeasible but has no IPR. We shall see that this is the only exception.

3. Irreducible instances. Let us consider the following three types of reductions:

(1) Let Z be a 2-cut and $Z \cap (X \cup Y) = \emptyset$. Let u be the tail of the arc from $V - Z$ to Z and v the head of the arc from Z to $V - Z$. Delete Z, and if $u \neq v$ then add the arc (u, v) to $G[V - Z]$. See Fig. 4(1).

(2) Let Z be a 2-cut, $|Z| \geq 2$, and $|Z \cap (X \cup Y)| = 1$. Then contract Z to the terminal $t \in Z$, deleting any resulting loops. (The resulting terminal t has degree 2.) See Fig. 4(2).

(3) Let Z be a 4-cut such that $G[Z]$ is connected, $|Z| \geq 2$, and $Z \cap (X \cup Y) = \emptyset$. Then contract Z into a single vertex. See Fig. 4(3).

The next lemma is immediate from the definition of reductions.

LEMMA 3.1. *An instance* $(G; X, Y)$ *is feasible if and only if it is feasible after performing any of the reductions* (1), (2), *or* (3). □

We say that an instance $(G; X, Y)$ is *reducible* if one of the above reductions (1)–(3) can be applied (in this case, we also call such 2-cuts or 4-cuts *reducible*); otherwise they are *irreducible*. An irreducible instance cannot have a 4-cut W with $W \cap (X \cup Y) = \emptyset$ even if W does not induce a connected subdigraph, because in such a case there is a 2-cut $Z \subset W$ with $Z \cap (X \cup Y) = \emptyset$ (i.e., it is reducible). As will be shown in section 9, an irreducible instance of a given instance $(G; X, Y)$ can be obtained in polynomial time. In this section, we present some properties of infeasible irreducible instances.

LEMMA 3.2. *Any minimal infeasible instance is irreducible.*

Proof. The proof is obvious because any of the reductions (1), (2), and (3) can be performed by an adequate sequence of arc contractions, during which any infeasible instance never becomes feasible by Lemma 3.1. □

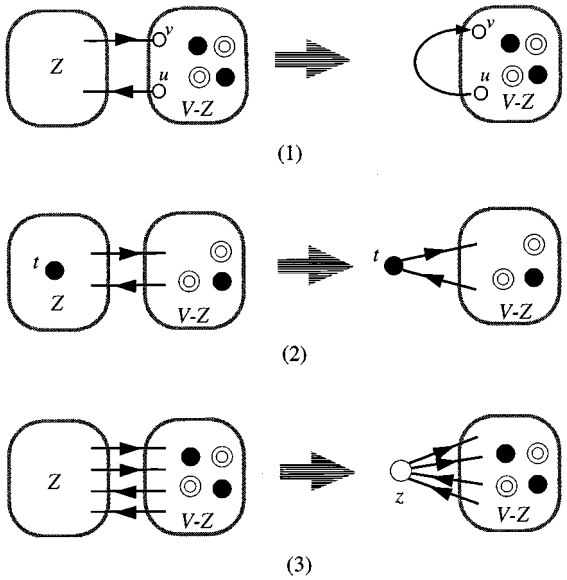


FIG. 4. *Three types of reductions of irreducible cuts Z.*

LEMMA 3.3. *Let $(G = (V, E); X, Y)$ be an infeasible irreducible instance. Then the following hold.*

(i) *There is no 2-cut Z that separates X and Y.*
(ii) *Each terminal has degree 2.*
(iii) *Each nonterminal vertex has degree 4.*
(iv) *For any pair of vertices $u, v \in V$, there is at most one arc connecting them (i.e., at most one of (u, v) and (v, u) exists in E).*

Proof. (i) Assume that a 2-cut Z separates X and Y, where $\delta(Z) = \{e_1, e_2\}$. Since G is connected and Eulerian, there are arc-disjoint x_1x_2-path P_X and x_2x_1-path P'_X. Clearly, one of them (say, P_X) contains no arcs from $\{e_1, e_2\}$. Similarly G has a $y'y''$-path P_Y, where $\{y', y''\} = Y$, such that $E(P_Y) \cap \{e_1, e_2\} = \emptyset$. These P_X and P_Y are arc-disjoint in G, and hence $(G; X, Y)$ is feasible.

(ii) Assume $deg(x_1) \geq 4$ for a terminal $x_1 \in X$ without loss of generality. If G has arc-disjoint x_1y_1-path P_1 and x_1y_2-path P_2, then $G - E(P_1) - E(P_2)$ has arc-disjoint y_1x_1-path P_3 and y_2x_1-path P_4, since G is Eulerian. Let H be the connected component in $G - E(P_1) - E(P_2) - E(P_3) - E(P_4)$ that contains x_2. Since G is connected, H must contain a vertex z in $V(P_i)$ for some i. Assume $z \in V(P_1) \cup V(P_2)$ (the case of $z \in V(P_3) \cup V(P_4)$ can be treated similarly). Then $E(P_1) \cup E(P_2) \cup E(H)$ contains a path P_X from x_1 to x_2 via z. However, y_1 and y_2 are connected in $G - E(P_1) - E(P_2) - E(H)$, since $Q_Y = \langle P_3, P_4 \rangle$ is a y_1y_2-chain, and the instance would be feasible by Lemma 2.1, contradicting the assumption. Therefore, at least one of the above P_1 and P_2 does not exist; i.e., by Menger's theorem, there must be a 2-cut W such that $x_1 \in W$ and $Y \subseteq V - W$. Since $deg(x_1) \geq 4$, we have $|W| \geq 2$. From (i) of this lemma, $x_2 \in V - W$ holds. This, however, implies that there is a reducible 2-cut W, which is a contradiction.

(iii) Assume $deg(u) \geq 6$ for a nonterminal vertex u. Let W be a cut that minimizes $|\delta(W)|$ among cuts W such that $u \in W$ and $\{x_1, y_1, y_2\} \subseteq V - W$. By the minimality of $|\delta(W)|$, $G[W]$ is connected. By $deg(u) \geq 6$, $|\delta(W)| = 2$ would imply $|W| \geq 2$, and W is reducible. Hence $|\delta(W)| \geq 4$. From this, either (a) $|\delta(W)| \geq 6$ or (b)

$|\delta(W)| = 4$ and $x_2 \in W$ must hold (otherwise, W would be reducible). In the case of (a), by Menger's theorem G has three arc-disjoint paths P_1, P_2, and P_3 from u to some vertices $w_1, w_2, w_3 \in \{x_1, y_1, y_2\}$. By (ii) of this lemma, every terminal has degree 2, and then we can assume that these paths P_1, P_2, and P_3 are ux_1-, uy_1-, and uy_2-paths, respectively. Since G is Eulerian, $G - E(P_1) - E(P_2) - E(P_3)$ has three arc-disjoint x_1u-path P_4, y_1u-path P_5, and y_2u-path P_6. Let H be the connected component in $G - \bigcup_{i=1,\dots,6} E(P_i)$ that contains x_2. Since G is connected, H must contain a vertex z in $V(P_i)$ for some i. Assume $z \in V(P_1) \cup V(P_4) \cup V(P_2) \cup V(P_3)$ (the case of $z \in V(P_1) \cup V(P_4) \cup V(P_5) \cup V(P_6)$ can be treated analogously). Then $E(P_1) \cup E(P_4) \cup E(P_2) \cup E(P_3) \cup E(H)$ contains a path P_X from x_1 to x_2 via z. However, y_1 and y_2 are connected in $G - E(P_1) - E(P_4) - E(P_2) - E(P_3) - E(H)$, since $Q_Y = \langle P_5, P_6 \rangle$ is a y_1y_2-chain, and the instance would be feasible by Lemma 2.1, contradicting the assumption.

Therefore, we assume (b); i.e., there is a 4-cut W separating $\{u, x_2\}$ and $\{x_1, y_1, y_2\}$. In this case, applying the above argument to u and $\{x_2, y_1, y_2\}$, we can conclude that there is also a 4-cut W' such that $\{u, x_1\} \subseteq W'$ and $\{x_2, y_1, y_2\} \subseteq V - W'$. These two cuts W and W' cross each other, and from (2.1) we have

$$|\delta(W)| + |\delta(W')| \geq |\delta(W \cap W')| + |\delta(W \cup W')|.$$

Here $|\delta(W)| = |\delta(W')| = 4$ and $|\delta(W \cup W')| \geq 4$ by (i). This implies $|\delta(W \cap W')| \leq 4$; i.e., $W \cap W'$ with $(W \cap W') \cap (X \cup Y) = \emptyset$ is a reducible 4-cut (or contains a reducible 2-cut $W'' \subset W \cap W'$), which is a contradiction.

(iv) If there are multiple arcs (u, v) and (u, v) in E, then u and v are nonterminal vertices by (iii) and $deg(u) = deg(v) = 4$ holds. This means that $Z = \{u, v\}$ is a reducible 2- or 4-cut, which is a contradiction. Similarly if there are two arcs (u, v) and (v, u), it is also easy to show that there is a reducible 2- or 4-cut $Z = \{u, v\}$, which is a contradiction. □

LEMMA 3.4. *Let $(G; X, Y)$ be an irreducible infeasible instance. Then G has at most two 2-cuts that separate $\{x_1, y'\}$ and $\{x_2, y''\}$, where $Y = \{y', y''\}$. If there are two such 2-cuts Z and Z', then G has a cut vertex z such that $Z = Z' \cup \{z\}$ or $Z = (V - Z') \cup \{z\}$. Conversely, any cut vertex is obtained in this manner.*

Proof. Let Z and Z' be the two 2-cuts that separate $\{x_1, y'\}$ and $\{x_2, y''\}$, where we assume $Z \cap (X \cup Y) = Z' \cap (X \cup Y) = \{x_1, y'\}$ without loss of generality. Choose Z (resp., Z') as the cut minimizing $|Z|$ (resp., maximizing $|Z'|$) among such 2-cuts. We first show that any other 2-cut Z'' that separates $\{x_1, y'\}$ and $\{x_2, y''\}$ satisfies

$$Z \subset Z'' \subset Z' \text{ (and hence } Z \subset Z'). \tag{3.1}$$

If Z'' crosses Z, we have $|\delta(Z \cap Z'')| \geq 4$ from the choice of Z and $|\delta(Z \cup Z'')| \geq 2$ as $Z \cup Z''$ separates $\{x_1, y'\}$ and $\{x_2, y''\}$. This means

$$|\delta(Z)| + |\delta(Z'')| = 4 < 6 \leq |\delta(Z \cap Z'')| + |\delta(Z \cup Z'')|,$$

which is a contradiction to (2.1). Similarly, we see that Z'' also cannot cross Z', and hence (3.1) holds. Since $|\delta(Z' - Z)| \leq |\delta(Z)| + |\delta(Z')|$ $(= 4)$, $Z' - Z$ is a 2- or 4-cut satisfying $(Z' - Z) \cap (X \cup Y) = \emptyset$. Then, by irreducibility, $Z' - Z$ must be a 4-cut consisting of a single vertex (say, z). This implies that z is a cut vertex. From $|Z' - Z| = 1$ and (3.1), $(G; X, Y)$ has at most two 2-cuts that separate $\{x_1, y'\}$ and $\{x_2, y''\}$.

Conversely, let z be a cut vertex in G. Since G is Eulerian, z cannot have degree 2. By Lemma 3.3(ii), (iii), and (iv), z is a nonterminal vertex and there are four distinct

vertices, say, u_1, u_2, v_1, v_2, adjacent to z. Again since G is Eulerian, $G - \{z\}$ has exactly two components. Let $W_1, W_2 \subset V$ be the vertex sets of these components, and assume without loss of generality that $(u_1, z), (u_2, z), (z, v_1), (z, v_2) \in E$, $u_1, v_1 \in W_1$ and $u_2, v_2 \in W_2$ since G is Eulerian and at least one arc is going out (resp., going in) of each W_1 and W_2. Clearly W_1 and $W_2 = V - (W_1 \cup \{z\})$ are both 2-cuts. By irreducibility and Lemma 3.3(i), these satisfy $|W_1 \cap X| = |W_1 \cap Y| = |(W_1 \cup \{z\}) \cap X| = |(W_1 \cup \{z\}) \cap Y| = 1$. This implies that a cut vertex z is obtained in the manner of the lemma statement. □

LEMMA 3.5. *Let $(G = (V, E); X, Y)$ be an irreducible infeasible instance which has an IPR, and let B be the cycle of the outer face in the IPR. If $|V| \geq 5$ and there is no 2-cut Z such that $|Z \cap X| = |Z \cap Y| = 1$ and $\min\{|Z|, |V - Z|\} \geq 3$, then $V - (X \cup Y)$ induces a connected digraph in $G - E(B)$.*

Proof. By $|V| \geq 5$, $V - (X \cup Y) \neq \emptyset$. By Lemma 3.3(ii) and the definition of IPR, each terminal in $X \cup Y$ is an isolated vertex in $G - E(B)$. Assume that $V - (X \cup Y)$ induces two connected components H_1 and H_2 in $G - E(B)$. By the planarity of the IPR, any vertices $u_1, v_1 \in V(H_1) \cap V(B)$ and any vertices $u_2, v_2 \in V(H_2) \cap V(B)$ cannot appear alternately (in the order of u_1, u_2, v_1, v_2) along cycle B. This means that there are two arcs $(u, v), (u', v') \in E(B)$ such that $V(H_1)$ and $V(H_2)$ are contained in two distinct connected components H_1' and H_2' in $G - \{(u, v), (u', v')\}$, respectively. Hence $Z = V(H_1')$ is a 2-cut, and by the irreducibility, both $V(H_1')$ and $V(H_2')$ must contain two terminals, one from X and the other from Y by Lemma 3.3(i). Clearly, each of $V(H_1')$ and $V(H_2')$ contains a nonterminal, and has at least three vertices; i.e., $\min\{|Z|, |V - Z|\} \geq 3$. □

The next lemma can be shown by inspecting all possible irreducible and infeasible instances with $|V| \leq 7$, based on Lemma 3.3.

LEMMA 3.6. *Let $(G; X, Y)$ be an irreducible infeasible instance with $|V| \leq 7$. If $|V| \in \{4, 5, 7\}$, then $(G; X, Y)$ has an IPR. If $|V| = 6$, $(G; X, Y)$ is the instance shown in Fig.* 3(c) *(in this case there is no irreducible infeasible instance with $|V| = 6$ in which some two terminals are adjacent to each other).* □

In this paper, we prove the next result.

THEOREM 3.7. *Let $(G; X, Y)$ be a minimal infeasible instance, and let it satisfy $|V| \neq 6$. Then $(G; X, Y)$ has an IPR.* □

We shall need sections 4–8 to prove Theorem 3.7 for general $|V| \geq 8$.

4. Outline of the proof. This section describes an outline of how to prove Theorem 3.7 in sections 5–8. We first assume that there is a smallest counterexample $(G^*; X, Y)$ to Theorem 3.7; i.e.,

$$(G^*; X, Y) \text{ is a minimal infeasible instance with } n \neq 6 \text{ vertices, but has no IPR,} \tag{4.1}$$

where G^* minimizes the number n^* of vertices among such instances. By Lemma 3.6, $n^* \geq 8$ is assumed. In sections 5 and 6, we characterize cut vertices, 2-cuts, and 6-cuts in G^*. Then in sections 7 and 8, as outlined below, we derive a contradiction from the existence of such G^*, which proves Theorem 3.7.

For the subsequent discussion, we introduce two operations. Let w be a nonterminal vertex with four incident arcs $(s_0, w), (s_1, w), (w, s_2), (w, s_3)$, where s_0, s_1, s_2, and s_3 are all distinct. We say that arcs (s_0, w) and (w, s_2) are *split off* at w when four arcs $(s_0, w), (s_1, w), (w, s_2), (w, s_3)$ are replaced with two new arcs (s_0, s_2) and (s_1, s_3) after eliminating w. Conversely, we say that two arcs $e = (u, v)$ and $e' = (u', v')$ are *hooked up* (with a new vertex w) when we replace these two arcs with the new arcs $(u, w), (w, v), (u', w)$, and (w, v') after introducing a new vertex w.

Now we choose a nonterminal vertex w adjacent to a terminal (say, x_2) by arc (x_2, w) in G^* and split off two arcs at w (recall that $deg(w) = 4$). If the resulting instance

$$(G^*_w; X, Y) \text{ remains connected and irreducible,}$$

then we call such splitting (or two arcs) *admissible*. Based on the properties obtained in sections 5 and 6, we show in section 7 that G^* always has an admissible splitting. Clearly

$$(G^*_w; X, Y) \text{ is infeasible}$$

since $(G^*; X, Y)$ is infeasible. Also if $(G^*_w; X, Y)$ is irreducible, then

$$(G^*_w; X, Y) \text{ has an IPR}$$

by the assumption on G^*.

However, we shall show in section 8 that, for the arc $e = (x_2, v)$ and any other arc e' in an irreducible infeasible instance $(G; X, Y)$ that has an IPR, the instance $(G_{e,e'}; X, Y)$ obtained by hooking up e and e' satisfies one of the following properties:

(i) $(G_{e,e'}; X, Y)$ is reducible,
(ii) $(G_{e,e'}; X, Y)$ has an IPR,
(iii) $(G_{e,e'}; X, Y)$ is feasible.

Notice that G^* is obtained from G^*_w by hooking up two arcs in an IPR of G^*_w. However, this leads to a contradiction because $G^* = (G^*_w)_{e,e'}$ satisfies none of (i)–(iii). Hence no such counterexample $(G^*; X, Y)$ exists.

5. Cut vertex and 2-cuts in G^*.

LEMMA 5.1. *The minimum counterexample* $(G^*; X, Y)$ *in* (4.1) *has the following properties:*

(i) *There is no cut vertex.*
(ii) *There is no* 2*-cut* Z *such that* $|Z \cap X| = |Z \cap Y| = 1$ *and* $\min\{|Z|, |V - Z|\} \geq 3$.

Proof. (i) Assume that $G^* = (V, E)$ has a cut vertex z, since no vertex with degree 2 is a cut vertex in a connected Eulerian digraph and z is nonterminal and has degree 4 by Lemma 3.3(ii), (iii).

Let Z' and Z'' be the vertex sets of the two components in $G^* - \{z\}$. By Lemma 3.4, $|Z' \cap X| = |Z' \cap Y| = 1$ holds and each of $Z' \cup \{z\}$ and $Z'' \cup \{z\}$ is a 2-cut in G^*. Let $(u', z), (u'', z), (z, v'), (z, v'') \in E$ be the four arcs incident to z. Without loss of generality we can assume that $u', v' \in Z'$ and $u'', v'' \in Z''$, $Z' \cap (X \cup Y) = \{x_1, y_2\}$, and G^* has an Eulerian cycle which visits terminals in the order of y_2, x_1, y_1, x_2 (if there is an Eulerian cycle in the order of y_2, x_1, x_2, y_1, then the instance is feasible). We decompose $(G^*; X, Y)$ into $(G'; X', Y')$ and $(G''; X'', Y'')$ as follows. Let $G^*[Z']$ (resp., $G^*[Z'']$) denote the subdigraph of G^* induced by Z' (resp., Z''), and let G' (resp., G'') be the Eulerian digraph obtained by adding new vertices y'_1, x'_2 and new arcs $(u', y'_1), (y'_1, x'_2), (x'_2, v')$ (resp., new vertices y''_2, x''_1 and new arcs $(u'', y''_2), (y''_2, x''_1), (x''_1, v'')$) to $G^*[Z']$ (resp., $G^*[Z'']$). Regard $X' = \{x_1, x'_2\}$, $Y' = \{y'_1, y_2\}$, $X'' = \{x''_1, x_2\}$, and $Y'' = \{y_1, y''_2\}$ as the sets of new terminals. We show that $(G'; X', Y')$ is irreducible. If $(G'; X', Y')$ has a reducible cut W, then W must separate y'_1 and x'_2 (otherwise W would be reducible in $(G^*; X, Y)$). Then W is a 2-cut such that $|W| \geq 2$ and $W \cap (X' \cup Y') = \{y'_1\}$ or $\{x'_2\}$. Since $deg(y'_1) = deg(x'_2) = 2$, $W - \{y'_1\}$ (or $W - \{x'_2\}$) is a 2-cut, which is reducible in $(G^*; X, Y)$, which is a contradiction. Note that $(G^*; X, Y)$ is infeasible only when both

new instances $(G'; X', Y')$ and $(G''; X'', Y'')$ are infeasible. Therefore, $(G'; X', Y')$ must be irreducible and infeasible. Clearly, instance $(G'; X', Y')$ is smaller than G^* (since $Z'' \cup \{z\}$ is replaced with two vertices in the new instance), and hence has an IPR by definition of G^* (note that $|V(G')| \neq 6$ by Lemma 3.6 since $(G'; X', Y')$ has two adjacent terminals). Analogously, we can show that $(G''; X'', Y'')$ also has an IPR. However, it is easy to see that G^* has an IPR if both instances $(G'; X', Y')$ and $(G''; X'', Y'')$ have IPRs, which is a contradiction.

(ii) Let Z be such a 2-cut in $(G^*; X, Y)$, where $\delta(Z) = \{(u', v''), (u'', v')\}$ and $u', v' \in Z$ and $u'', v'' \in V - Z$. Clearly, $\overline{Z} = V - Z$ is also such a 2-cut. Note that u' and v' (resp., u'' and v'') are distinct (otherwise it would be a cut vertex, contradicting the above (i)). Without loss of generality assume that $Z \cap (X \cup Y) = \{x_1, y_2\}$ and G^* has an Eulerian cycle that visits terminals in the order of y_2, x_1, y_1, x_2. We decompose instance $(G^*; X, Y)$ into the two instances $(G'; X', Y')$ and $(G''; X'', Y'')$ as follows. Let G' (resp., G'') be the digraph obtained by adding new vertices y_1', x_2' and new arcs $(u', y_1'), (y_1', x_2'), (x_2', v')$ (resp., new vertices y_2'', x_1'' and new arcs $(u'', y_2''), (y_2'', x_1''), (x_1'', v''))$ to $G^*[Z]$ (resp., $G^*[\overline{Z}]$). Regard $X' = \{x_1, x_2'\}$, $Y' = \{y_1', y_2\}$, $X'' = \{x_1'', x_2\}$, and $Y'' = \{y_1, y_2''\}$ as the sets of new terminals. Analogously to (i), we see that each of the new instances is an irreducible infeasible instance. From the assumption of $\min\{|Z|, |V - Z|\} \geq 3$, each of the new instances is smaller than G^* and has an IPR by definition of G^* or by Lemma 3.6. However, it is again clear that G^* has an IPR if these new instances have IPRs, which is a contradiction. ∎

6. 6-cuts. We first observe a property of a 6-cut Z.

LEMMA 6.1. *Let $(G = (V, E); X, Y)$ be an infeasible instance. If there exists a 6-cut Z with $Z \cap (X \cup Y) = \emptyset$ satisfying the following* (i)–(iv)*, then $(G; X, Y)$ is irreducible.*

(i) $|Z| \geq 3$.
(ii) *Any cut W with $W \subseteq Z$ is irreducible.*
(iii) *Any cut W with $W \supseteq Z$ or $W \cap Z = \emptyset$ is irreducible.*
(iv) *$\delta(Z)$ contains no multiple arcs.*

Proof. Let Z be such a 6-cut. From (ii) and (iii), it suffices to show that any cut W which intersects Z is irreducible; i.e., $|\delta(W)| = 2$ or 4. Assume that a cut W intersecting Z is reducible. Since W contains at most one terminal, $V - (W \cup Z)$ contains a terminal, and hence cuts W and Z cross each other. By (2.1),

$$|\delta(W)| + |\delta(Z)| \geq |\delta(W \cap Z)| + |\delta(W \cup Z)|, \tag{6.1}$$

and by (2.2),

$$|\delta(W)| + |\delta(Z)| = |\delta(W - Z)| + |\delta(Z - W)| + 2|\delta(W \cap Z, V - (W \cup Z))|. \tag{6.2}$$

Since Z contains no reducible cut by (ii), we have $|\delta(W \cap Z)| \geq 4$. Also by (iii), we see that $|\delta(W \cup Z)| \geq |\delta(W)| + 2$ holds (otherwise if $|\delta(W \cup Z)| \leq |\delta(W)|$, then $W \cup Z$ would be reducible by $(W \cup Z) \cap (X \cup Y) = W \cap (X \cup Y)$). Therefore, $|\delta(Z)| = 6$, $|\delta(W \cap Z)| \geq 4$, and $|\delta(W \cup Z)| \geq |\delta(W)| + 2$ imply that (6.1) holds by equality. Hence $|\delta(W \cap Z)| = 4$ holds, and this means $|W \cap Z| = 1$ since Z contains no reducible cut. Then $|Z - W| \geq 2$ by (i), and again by (ii), $|\delta(Z - W)| \geq 6$. By (iii), we have $|\delta(W - Z)| \geq |\delta(W)|$ (otherwise, if $|\delta(W - Z)| \leq |\delta(W)| - 2$, then $W - Z$ would be reducible). By $|\delta(Z - W)| \geq 6$, $|\delta(Z)| = 6$, and $|\delta(W - Z)| \geq |\delta(W)|$, (6.2) implies that $|\delta(Z - W)| = |\delta(Z)| = 6$, $|\delta(W - Z)| = |\delta(W)|$, and $|\delta(W \cap Z, V - (W \cup Z))| = 0$. By (iii), $|W - Z| = 1$ must hold, since $|W - Z| \geq 2$ and $|\delta(W - Z)| = |\delta(W)|$ mean

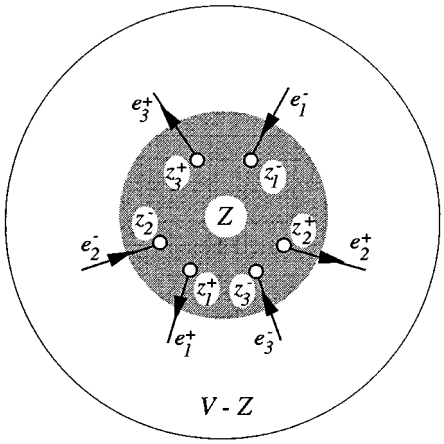


FIG. 5. *A 6-cut Z.*

that $W - Z$ is reducible. Now the vertex $v \in W \cap Z$ has degree 4 and has no adjacent vertex in $V - (W \cup Z)$ by $|\delta(W \cap Z, V - (W \cup Z))| = 0$. From $|\delta(Z - W)| = |\delta(Z)|$ and $|\delta(W - Z)| = |\delta(W)|$, we then have $|\delta(\{v\}, W - Z)| = |\delta(\{v\}, Z - W)| = 2$. However, by $|W - Z| = 1$, the two arcs in $\delta(\{v\}, Z - W)$ are multiple, contradicting (iv). □

6.1. Interchangeability. Let Z be a 6-cut in $(G; X, Y)$ such that

$$Z \cap (X \cup Y) = \emptyset,$$

$$\delta^-(Z) = \{e_1^-, e_2^-, e_3^-\}, \quad \delta^+(Z) = \{e_1^+, e_2^+, e_3^+\}, \tag{6.3}$$

where z_i^- (resp., z_i^+) denote the head vertices of arcs e_i^- (resp., the tail vertices of arcs e_i^+) for $i = 1, 2, 3$ (see Fig. 5). Note that these vertices z_i^- and z_i^+ may not be distinct. We say that Z is $(e_1^-, e_2^-, e_3^-; e_{j_1}^+, e_{j_2}^+, e_{j_3}^+)$-*interchangeable*, where $\{j_1, j_2, j_3\} = \{1, 2, 3\}$, if the subdigraph $G[Z]$ of G induced by Z has three arc-disjoint $z_i^- z_{j_i}^+$-paths P_{i,j_i} $(i = 1, 2, 3)$. Z is called *fully interchangeable* if it is $(e_1^-, e_2^-, e_3^-; e_{j_1}^+, e_{j_2}^+, e_{j_3}^+)$-interchangeable for any choice of j_1, j_2, j_3 with $\{j_1, j_2, j_3\} = \{1, 2, 3\}$.

LEMMA 6.2. *An irreducible infeasible instance $(G; X, Y)$ has no fully interchangeable 6-cut Z with $Z \cap (X \cup Y) = \emptyset$.*

Proof. Assume that there is such a 6-cut Z, and let G_Z be the digraph obtained by contracting Z into a nonterminal vertex z. It is easy to see the following:

(i) $(G; X, Y)$ is feasible if and only if $(G_Z; X, Y)$ is feasible, and

(ii) $(G_Z; X, Y)$ is irreducible.

Therefore, $(G_Z; X, Y)$ is also an irreducible infeasible instance, but $deg(z) = 6$ contradicts Lemma 3.3(iii). □

A directed cycle of length 3 is called a *triangle*.

LEMMA 6.3. *Let $(G; X, Y)$ be an irreducible infeasible instance. Then the following hold.*

(i) *If Z is a 6-cut such that $Z \cap (X \cup Y) = \emptyset$, $|Z| = 3$, and the induced subdigraph $G[Z]$ is connected, then $G[Z]$ is a triangle.*

(ii) *If $|Z| = 3$ and the three vertices in Z are mutually adjacent, then the induced subdigraph $G[Z]$ is a triangle.*

Proof. (i) From Lemma 3.3(iv) and $|\delta(Z)| = 6$, it is easy to see that the connected subdigraph $G[Z]$ contains exactly three arcs and these three arcs form an undirected

cycle C of length 3 if the orientation is neglected. This C must be a directed cycle in G, because otherwise it is not difficult to see, by checking all possibilities, that Z is fully interchangeable, contradicting Lemma 6.2.

(ii) If all vertices in Z are nonterminal, (ii) follows from (i). Therefore, assume that Z contains a terminal. $Z = \{v_1, v_2, v_3\}$ can contain at most two terminals since any terminal has degree 2 from Lemma 3.3(ii). If Z contains two terminals, then clearly $G[Z]$ forms a triangle. Then assume that Z contains exactly one terminal, say, $Z \cap (X \cup Y) = \{v_2\}$. If $G[Z]$ is not a triangle, then we can assume without loss of generality that $G[Z]$ has arcs $(v_1, v_2), (v_2, v_3), (v_1, v_3)$. Let G_{v_2} be the digraph obtained from G by contracting Z into terminal v_2. It is easy to see that $(G_{v_2}; X, Y)$ is also an irreducible infeasible instance. But v_2 has degree 4 and contradicts Lemma 3.3(ii). □

Let $\delta(Z; G)$ denote $\delta(Z)$ in a digraph G.

LEMMA 6.4. *Let $(G; X, Y)$ be an irreducible infeasible instance, and let Z be a* 6*-cut in $(G; X, Y)$ as defined in* (6.3). *If Z is not $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$-interchangeable, then properties* (i)–(iv) *hold.*

(i) *The induced subdigraph $G[Z]$ is connected.*

(ii) *If $G[Z]$ has no $z_i^- z_i^+$-path for some $i \in \{1, 2, 3\}$, then $Z = \{z_i^-, z_i^+\}$.*

(iii) *$z_i^- \neq z_i^+$ for all $i \in \{1, 2, 3\}$.*

(iv) *If $|Z| \geq 3$, then $z_i^- \neq z_{i'}^-$ and $z_i^+ \neq z_{i'}^+$ for $1 \leq i < i' \leq 3$.*

Proof. Note that $|\delta^-(\{u\}; G[Z])| = |\delta^+(\{u\}; G[Z])|$ for all $u \in Z - \{z_i^-, z_i^+ \mid i = 1, 2, 3\}$. Hence if $G[Z]$ has a $z_i^- z_j^+$-path P, then $G[Z] - E(P)$ has arc-disjoint $z_{i'}^- z_{j'}^+$- and $z_{i''}^- z_{j''}^+$-paths for some i', i'', j', j'' with $\{i', i''\} = \{1, 2, 3\} - \{i\}$ and $\{j', j''\} = \{1, 2, 3\} - \{j\}$.

(i) If the subdigraph $G[Z]$ of G consists of more than one connected component, then there would be a reducible 2-cut Z' with $Z' \subset Z$.

(ii) Assume without loss of generality that $G[Z]$ has no $z_1^- z_1^+$-path. Then, by Menger's theorem, $G[Z]$ has a cut $W \subset Z$ such that $z_1^- \in W$, $z_1^+ \in Z - W$, and $|\delta^+(W; G[Z])| = 0$. Here $|\delta^-(W; G[Z])| \geq 1$ since $G[Z]$ is connected by (i). Let H denote the Eulerian digraph obtained by adding three new arcs $e_i^* = (z_i^+, z_i^-)$ $(i = 1, 2, 3)$ to $G[Z]$. Now $e_1^* \in \delta^-(W; H)$ and $|\delta^-(W; H)| \geq 2$ hold, and hence $|\delta^+(W; H)| \geq 2$ since H is Eulerian. Since $|\delta^+(W; G[Z])| = 0$, we see that $e_2^*, e_3^* \in \delta^+(W; H)$. Therefore, by $|\delta^+(W; H)| = 2 = |\delta^-(W; H)|$, we have $|\delta^-(W; G[Z])| = 1$. This implies that $z_1^-, z_2^+, z_3^+ \in W$ and $z_1^+, z_2^-, z_3^- \in Z - W$ and that $|\delta(W; G)| = |\delta(Z - W; G)| = 4$ holds. Hence the 4-cut W (resp., $Z - W$) in G consists of a single vertex z_1^- (resp., z_1^+), respectively, from the irreducibility of G.

(iii) If $z_1^- = z_1^+$, then $G[Z]$ has a $z_1^- z_1^+$-path of null length. Since G is Eulerian, $G[Z]$ has two arc-disjoint $z_2^- z_2^+$- and $z_3^- z_3^+$-paths, because even if $G[Z]$ has arc-disjoint $z_2^- z_3^+$- and $z_3^- z_2^+$-paths, the connectivity of $G[Z]$ (which follows from (i)) implies that these paths have a common vertex v from which $z_2^- z_2^+$- and $z_3^- z_3^+$-paths can be constructed. This contradicts that Z is not $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$-interchangeable.

(iv) From (ii), $|Z| \geq 3$ means that $G[Z]$ has paths from z_i^- to z_i^+ for all $i = 1, 2, 3$ (but they may not be arc-disjoint). Assume $z_1^- = z_2^-$ since other cases are analogous, and choose a $z_3^- z_3^+$-path P_3 in $G[Z]$. Note that $G[Z] - E(P_3)$ together with additional arcs (z_1^+, z_1^-) and (z_2^+, z_2^-) becomes Eulerian. This means that $G[Z] - E(P_3)$ has arc-disjoint $z_1^- z_1^+$- and $z_2^- z_2^+$-paths, where $z_1^- = z_2^-$. This contradicts that Z is not $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$-interchangeable. □

6.2. Proper 6-cuts in G^*. We call a 6-cut Z *proper* if

(a) $|Z| \geq 3$,

(b) $Z \cap (X \cup Y) = \emptyset$,

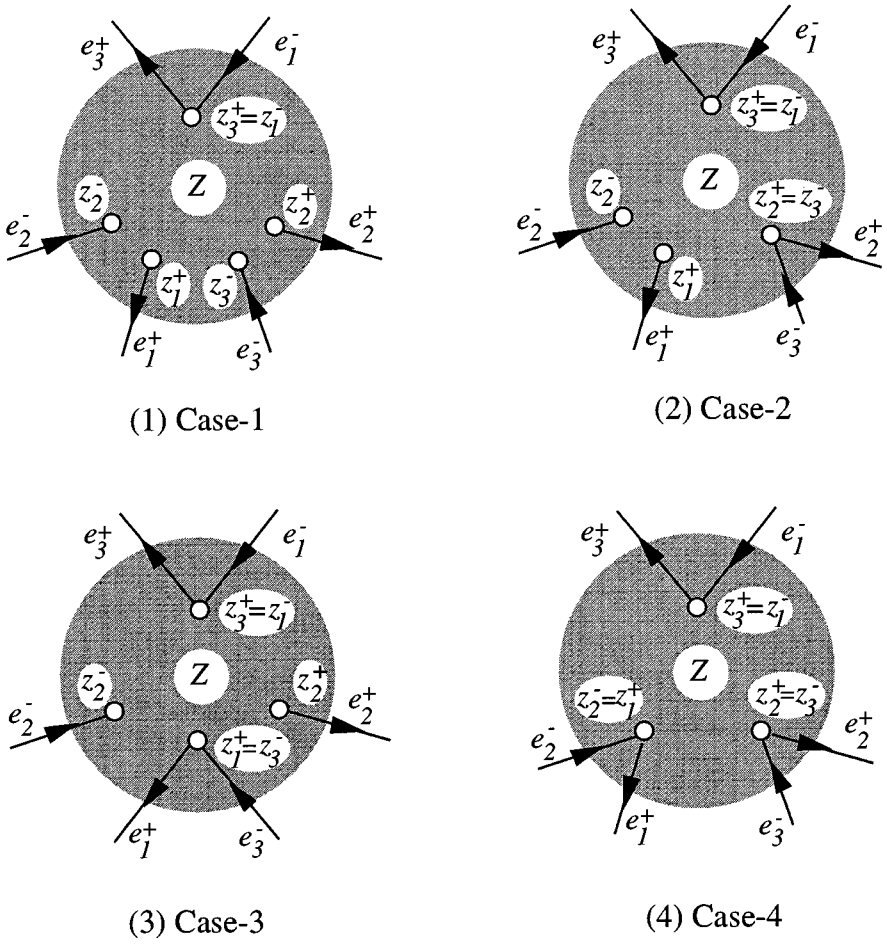


FIG. 6. *Four possible cases for a proper* 6*-cut* Z *in* $(G^*; X, Y)$.

(c) Z contains a vertex z such that $(u, z) \in \delta^-(Z)$ and $(z, v) \in \delta^+(Z)$.

In this subsection, we prove that any proper 6-cut Z induces a triangle in the minimum counterexample $(G^*; X, Y)$.

Let Z be a proper 6-cut in the minimum counterexample $(G^*; X, Y)$ for which $e_i^+, e_i^-, z_i^+, z_i^-$ $(i = 1, 2, 3)$ are defined by (6.3). As Z is not fully interchangeable by Lemma 6.2, assume that Z is not $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$-interchangeable without loss of generality. From condition (c) and Lemma 6.4(iii), $z_i^- = z_j^+$ holds for some $i \neq j$. Here we assume without loss of generality that $z_1^- = z_3^+$ (if necessary, exchange the indices $i = 2, 3$). By Lemma 6.4(iii) and (iv), we have the following four possible cases.

Case 1. $z_2^-, z_3^-, z_2^+, z_3^+$ are all distinct (see Fig. 6(1)).

Case 2. $z_2^+ = z_3^-$ and $z_2^- \neq z_1^+$ (or symmetrically, $z_2^- = z_1^+$ and $z_2^+ \neq z_3^-$) (see Fig. 6(2)).

Case 3. $z_1^+ = z_3^-$ and $z_2^- \neq z_2^+$ (see Fig. 6(3)).

Case 4. $z_2^+ = z_3^-$ and $z_2^- = z_1^+$ (see Fig. 6(4)).

Now let H_Z^* be the Eulerian digraph obtained from $G^*[Z]$ as follows (see Fig. 7):

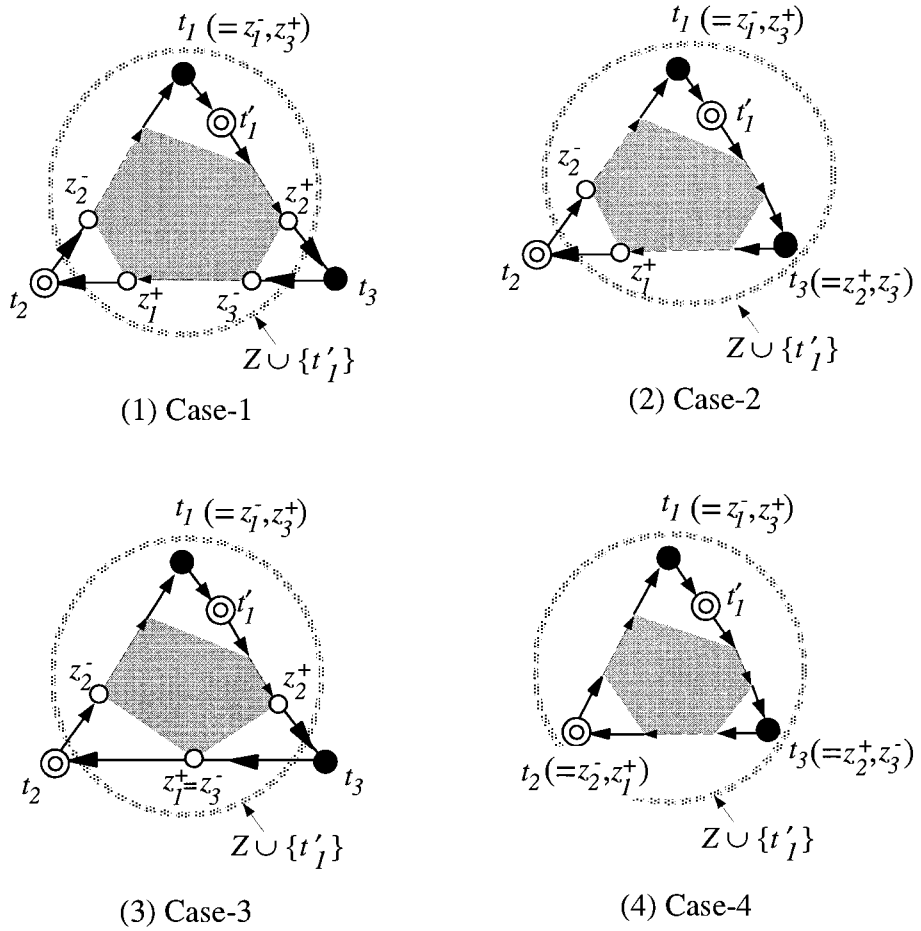


FIG. 7. *The instance* $(H_Z^*; \tilde{X}, \tilde{Y})$ *transformed from* $G^*[Z]$.

1. For $i = 1, 2$, if z_i^+ and z_{i+1}^- are distinct, add a new vertex t_{i+1} together with new arcs (z_i^+, t_{i+1}) and (t_{i+1}, z_{i+1}^-); if $z_i^+ = z_{i+1}^-$, then let $t_{i+1} = z_{i+1}^-(= z_i^+)$.
2. Let $t_1 = z_1^-(= z_3^+)$.
3. Replace the arc (t_1, v) with two arcs (t_1, t_1') and (t_1', v), inserting a new vertex t_1' between t_1 and v.

Define sets of terminals $\tilde{X} = \{t_1, t_3\}$ and $\tilde{Y} = \{t_1', t_2\}$. Obviously, instance $(H_Z^*; \tilde{X}, \tilde{Y})$ is feasible if and only if Z is $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$-interchangeable. Thus, $(H_Z^*; \tilde{X}, \tilde{Y})$ must be infeasible. Note that H_Z^* contains at most $|Z| + 3$ $(< |Z| + |X \cup Y| \leq n^*)$ vertices, where n^* is the number of vertices in G^*. The next lemma summarizes the properties of H_Z^*.

LEMMA 6.5. *Let* $(G^*; X, Y)$ *be the minimum counterexample, and let* Z *be a proper* 6*-cut in* $(G^*; X, Y)$*, which is not* $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$*-interchangeable, and* $(H_Z^*; \tilde{X}, \tilde{Y})$ *be the instance defined in the above. Then the following properties* (i)–(iv) *hold in all the above cases.*

(i) $(H_Z^*; \tilde{X}, \tilde{Y})$ *is infeasible.*

(ii) $(H_Z^*; \tilde{X}, \tilde{Y})$ *is connected and irreducible and has no* 2*-cut* W *such that* $|W \cap \tilde{X}| = |W \cap \tilde{Y}| = 1$ *and* $\min\{|W|, |V(H_Z^*) - W|\} \geq 3$.

(iii) $(H_Z^*; \tilde{X}, \tilde{Y})$ *has an IPR.*

(iv) *Z is $(e_1^-, e_2^-, e_3^-; e_{j_1}^+, e_{j_2}^+, e_{j_3}^+)$-interchangeable for any choice of j_1, j_2, j_3 from $\{1,2,3\}$ except $(j_1, j_2, j_3) = (1,2,3)$.*

Proof. In what follows, we consider all four cases simultaneously.

(i) Already proved.

(ii) Since $G^*[Z]$ is connected by Lemma 6.4(i), H_Z^* is connected. Assume that $(H_Z^*; \tilde{X}, \tilde{Y})$ has a reducible cut W. If $W \cap (\tilde{X} \cup \tilde{Y}) = \emptyset$, then W would also be a reducible cut in $(G^*; X, Y)$, which is a contradiction. Therefore, W must be a 2-cut in $(H_Z^*; \tilde{X}, \tilde{Y})$ such that $|W| \geq 2$ and $|W \cap (\tilde{X} \cup \tilde{Y})| = 1$. Since no such W with $|W| = 2$ attains $|\delta(W; H_Z^*)| = 2$ as easily checked, we further assume $|W| \geq 3$. Let $\{t^*\} = W \cap (\tilde{X} \cup \tilde{Y})$. Clearly, $|\delta(W; H_Z^*)| = 2$ implies that $|\delta(W'; G^*)| \leq 4$ holds for $W' = W - \{t^*\} \subseteq V$. This and $|W'| = |W| - 1 \geq 2$ mean that W' is a reducible 2- or 4-cut in $(G^*; X, Y)$, which is a contradiction. Therefore, $(H_Z^*; \tilde{X}, \tilde{Y})$ is irreducible. Assume that $(H_Z^*; \tilde{X}, \tilde{Y})$ has a 2-cut W with $|W \cap \tilde{X}| = |W \cap \tilde{Y}| = 1$ and $\min\{|W|, |V(H_Z^*) - W|\} \geq 3$, and let $t_1' \in W$ without loss of generality. Then $|W| \geq 3$ implies that $(W - (\tilde{X} \cup \tilde{Y})) \cup \{z_3^+\}$ is a reducible 4-cut in G^*, contradicting irreducibility of $(G^*; X, Y)$.

(iii) The instance $(H_Z^*; \tilde{X}, \tilde{Y})$ is infeasible by (i) and is connected and irreducible by (ii). Since H_Z^* contains at most $|Z| + 3 < n^*$ vertices, the instance $(H_Z^*; \tilde{X}, \tilde{Y})$ has an IPR by the minimality assumption on G^* and by Lemma 3.6 (note that terminals $t_1 \in \tilde{X}$ and $t_1' \in \tilde{Y}$ are adjacent).

(iv) Let B be the directed cycle of the outer face in an IPR of $(H_Z^*; \tilde{X}, \tilde{Y})$, where B visits t_1, t_1', t_3, t_2 in this order, and let $B(u, v)$ denote the uv-path on B, where $B(u, u)$ means a path of null length. Note that $(j_1, j_2, j_3) \neq (1,2,3)$ implies (a) $j_1 = 3$, (b) $j_1 = 2$, or (c) $j_1 = 1$ and $j_2 = 3$. If $|V(H_Z^*)| = 4$, then only Case 4 can occur and the IPR is a cycle of length 4 visiting t_1, t_1', t_3, t_2 in this order. In this case, we can easily check by inspection that (iv) holds. We then assume $|V(H_Z^*)| \geq 5$. Since $(H_Z^*; \tilde{X}, \tilde{Y})$ has no 2-cut W stated in the above (ii) and $|V(H_Z^*)| \geq 5$ holds, $V(H_Z^*) - (\tilde{X} \cup \tilde{Y})$ induces a connected component in $H_Z^* - E(B)$ by Lemma 3.5.

(a) $j_1 = 3$. We first take a $z_1^- z_3^+$-path P_A of null length in H_Z^*. We then consider path $B(z_2^+, z_2^-)$, which contains a $z_3^- z_1^+$-path $P_B = B(z_3^- z_1^+)$, and remove the arcs of $E(B(z_2^+, z_2^-))$ from H_Z^*. Now $indeg(u) = outdeg(u)$ holds for all $u \in V(H_Z^*) - \{z_2^+, z_2^-\}$. Then, the set $E(H_Z^*) - E(B(z_2^+, z_2^-))$ of remaining arcs can be regarded as a $z_2^- z_2^+$-path P_C. Therefore, Z is $(e_1^-, e_2^-, e_3^-; e_3^+, e_2^+, e_1^+)$-interchangeable. To show the $(e_1^-, e_2^-, e_3^-; e_3^+, e_1^+, e_2^+)$-interchangeability, it suffices to prove that the above $z_3^- z_1^+$-path $P_B = B(z_3^- z_1^+)$ and $z_2^- z_2^+$-path P_C have a common vertex by which we can reconstruct arc-disjoint $z_3^- z_2^+$-path and $z_2^- z_1^+$-path. Now since $V(H_Z^*) - (\tilde{X} \cup \tilde{Y})$ induces a connected component in $H_Z^* - E(B)$, we obtain $V(P_B) \cap V(P_C) \neq \emptyset$.

(b) $j_1 = 2$. It is easy to see that $z_1^- z_2^+$-path $P_A = B(z_1^-, z_2^+)$ and path $B(z_2^+, z_2^-)$ (which contains a $z_3^- z_1^+$-path $P_B = B(z_3^- z_1^+)$) and $z_2^- z_3^+$-path $P_C = E(H_Z^*) - E(B(z_1^-, z_2^-))$ are arc-disjoint. Therefore, Z is $(e_1^-, e_2^-, e_3^-; e_2^+, e_3^+, e_1^+)$-interchangeable. By the connectedness of $V(H_Z^*) - (\tilde{X} \cup \tilde{Y})$ in $H_Z^* - E(B)$, $V(P_B) \cap V(P_C) \neq \emptyset$ holds and arc-disjoint $z_3^- z_3^+$-path and $z_2^- z_1^+$-path can be reconstructed from P_B and P_C, implying $(e_1^-, e_2^-, e_3^-; e_2^+, e_1^+, e_3^+)$-interchangeability.

(c) $j_1 = 1$ and $j_2 = 3$. Consider a $z_2^- z_3^+$-path $P_A = B(z_2^-, z_3^+)$ path $B(z_2^+, z_2^-)$ (which contains a $z_3^- z_1^+$-path $P_B = B(z_3^- z_1^+)$), and $z_1^- z_2^+$-path $P_C = E(H_Z^*) - E(B(z_2^+, z_1^-))$. These three paths are arc-disjoint. By the connectedness of $V(H_Z^*) - (\tilde{X} \cup \tilde{Y})$ in $H_Z^* - E(B)$, $V(P_B) \cap V(P_C) \neq \emptyset$ holds and arc-disjoint $z_3^- z_2^+$-path and $z_1^- z_1^+$-path can be reconstructed from P_B and P_C, implying $(e_1^-, e_2^-, e_3^-; e_1^+, e_3^+, e_2^+)$-interchangeability. ∎

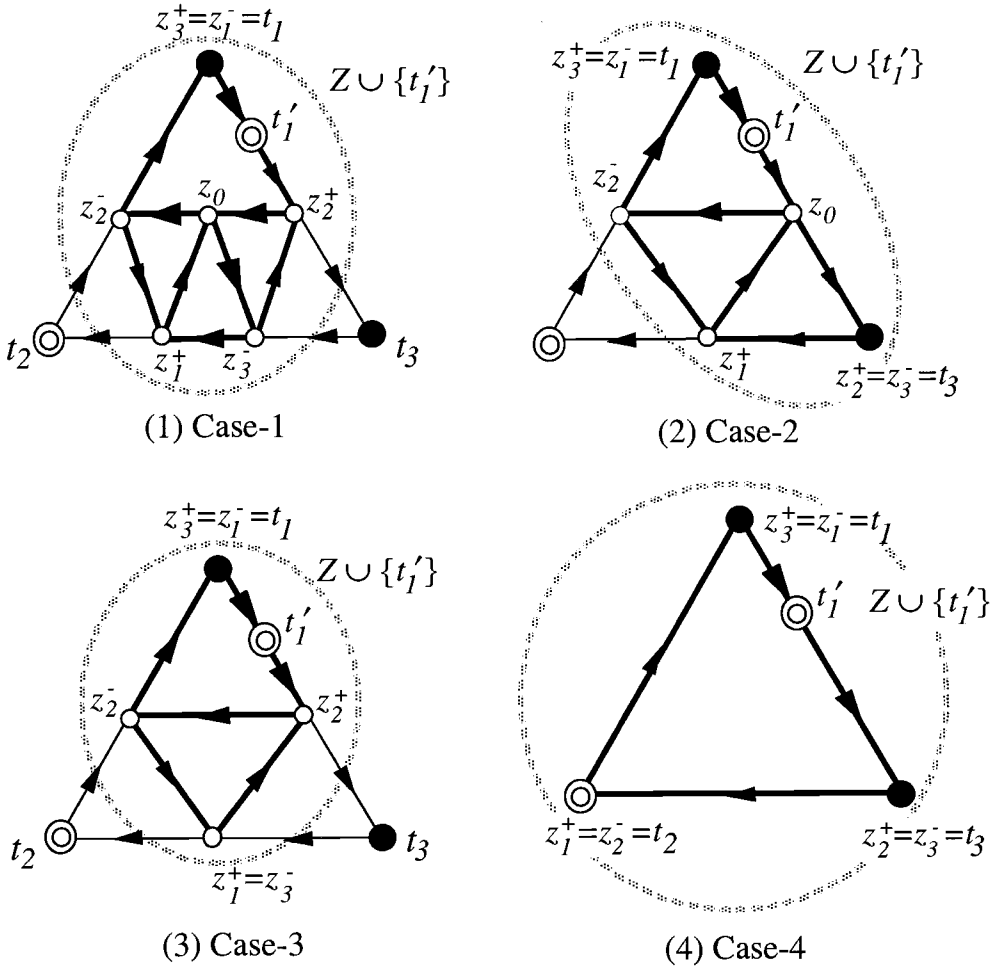


FIG. 8. *The smallest IPRs of* H_Z^*.

Figure 8 shows the smallest IPR of H_Z^* in Cases 1–4. Let $H_Z^*[Z \cup \{t'\}]$ be the subdigraph induced by $Z \cup \{t'\}$ from the smallest IPR of H_Z^* in Fig. 8, and let $H_Z^\#$ be the digraph obtained from $H_Z^*[Z \cup \{t'\}]$ by deleting the vertex t_1' (merging the two arcs $(z_3^+, t_1'), (t_1', v)$ into an arc (z_3^+, v)). Let us consider the digraph $G_{H_Z}^\#$ obtained from the minimum counterexample G^* by replacing $G^*[Z]$ by $H_Z^\#$, as shown in Fig. 9. Let $Z_0 = V(H_Z^\#)$. Let Z_0' be the set of vertices $u \in Z_0$ with $|\delta(\{u\}; H_Z^\#)| = 2$ and $z_0 \in Z_0$ be the vertex with $|\delta(\{u\}; H_Z^\#)| = 4$ in Cases 1 and 2. Note that each vertex in Z_0' is either z_i^+ or z_i^- for some i.

LEMMA 6.6. *For a proper* 6*-cut* Z *in the minimum counterexample* $(G^*; X, Y)$*, which is not* $(e_1^-, e_2^-, e_3^-; e_1^+, e_2^+, e_3^+)$*-interchangeable, let* $G_Z^\#$ *be defined as above. Then the following properties* (i)–(iv) *hold in all Cases* 1–4 *(defined in the beginning of this subsection).*

(i) $(G_Z^\#; X, Y)$ *is infeasible.*

(ii) $(G_Z^\#; X, Y)$ *is connected and irreducible.*

(iii) $(G^*; X, Y)$ *has an IPR if* $(G_Z^\#; X, Y)$ *has an IPR.*

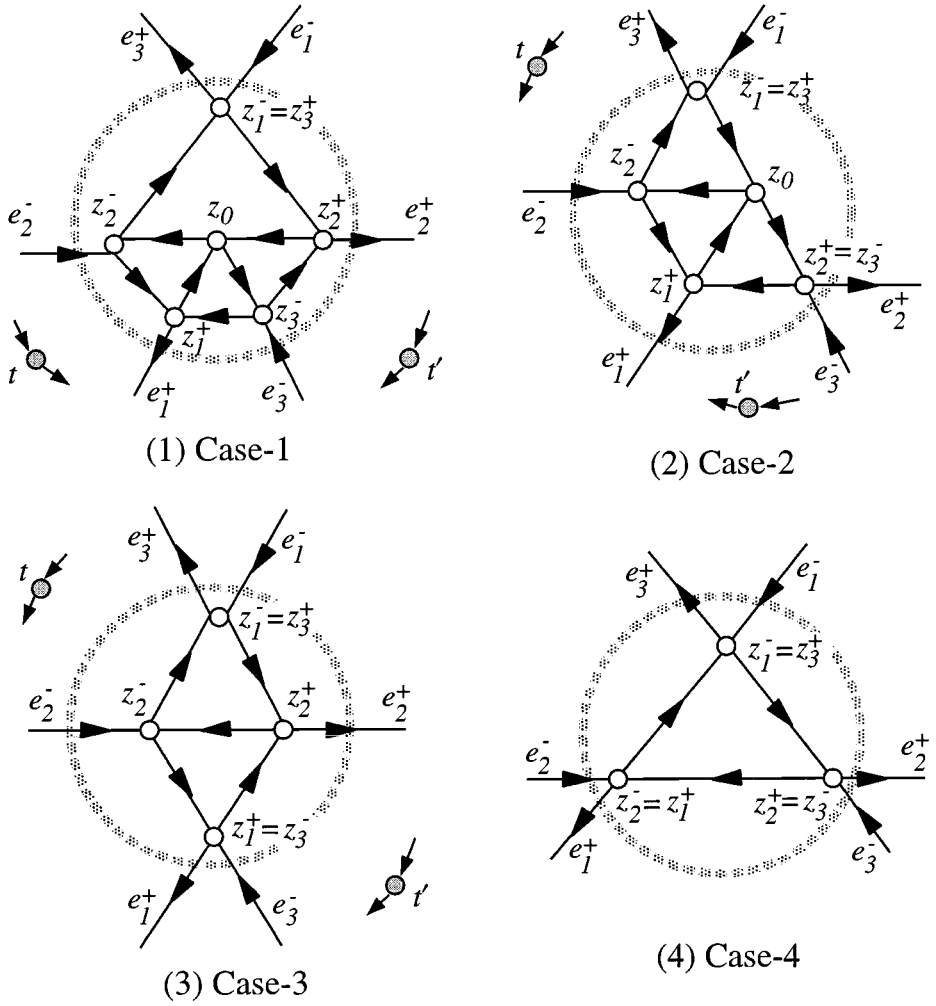


FIG. 9. *The instance $G^{\#}_Z$ obtained from G^* by replacing $G^*[Z]$ with $H^{\#}_Z$.*

(iv) $G^*[Z] = H^{\#}_Z$ *holds.*

Proof. (i) By Lemma 6.5(iv), the 6-cut Z in G^* is $(e_1^-, e_2^-, e_3^-; e_{j_1}^+, e_{j_2}^+, e_{j_3}^+)$-interchangeable for any choice of j_1, j_2, j_3 from $\{1, 2, 3\}$, except $(j_1, j_2, j_3) = (1, 2, 3)$. Then it is easy to see that the corresponding 6-cut $Z_0 = V(H^{\#}_Z)$ also has the same interchangeability in $G^{\#}_Z$, implying that $(G^{\#}_Z; X, Y)$ is feasible if and only if $(G^*; X, Y)$ is feasible. Since $(G^*; X, Y)$ is infeasible, $(G^{\#}_Z; X, Y)$ is also infeasible.

(ii) Clearly, $(G^{\#}_Z; X, Y)$ is connected since $(G^*; X, Y)$ is. We apply Lemma 6.1 to $(G^{\#}_Z; X, Y)$ and 6-cut $Z_0 = V(H^{\#}_Z)$. Clearly, $Z_0 \cap (X \cup Y) = \emptyset$ and $|Z_0| \geq 3$, and $\delta(Z_0)$ contains no multiple arcs, satisfying conditions (i) and (iv) of Lemma 6.1. We see by inspection that there is no reducible cut $W \subseteq Z_0$ and from the irreducibility of $(G^*; X, Y)$ that there is no reducible cut W with $W \supseteq Z_0$ or $W \cap Z_0 = \emptyset$, satisfying conditions (ii) and (iii) of Lemma 6.1. Therefore, $(G^{\#}_Z; X, Y)$ is irreducible by Lemma 6.1.

(iii) Consider Case 1 (other cases can be treated analogously). Assume that $(G^{\#}_Z; X, Y)$ has an IPR in which we assume without loss of generality that arcs

$e_3^+, e_1^-, e_2^+, e_3^-, e_1^+, e_2^-$ appear in this order along the cycle $\{z_3^+ = z_1^-, z_2^+, z_3^-, z_1^+, z_2^-\}$ (recall that, in an IPR, the arcs incident to each vertex are alternately oriented out and in). By Lemma 6.5(iii), $(H_Z^*; \tilde{X}, \tilde{Y})$ has an IPR in which we can assume without loss of generality that all terminals t_1, t_1', t_3, t_2 appear in this order along the cycle of the outer face, and hence the arcs $e_3^+, e_1^-, e_2^+, e_3^-, e_1^+, e_2^-$ appear in the same way as in the IPR of $(G_Z^\#; X, Y)$. This implies that $H_Z^\#$ in $G_Z^\#$ can be replaced with H_Z^* so that the resulting digraph G^* also has an IPR.

(iv) From (i)–(iii) and the assumption on G^*, $(G_Z^\#; X, Y)$ is an irreducible infeasible instance, but has no IPR. Clearly, by $|V(G^\#)| \geq |Z_0| + |X \cup Y| \geq 7$, $(G_Z^\#; X, Y)$ is also a counterexample to Theorem 3.7. Then by the minimality of G^*, $|Z_0| = |Z|$. By inspection, we see that Z with $|Z| = |Z_0|$ can induce no other subdigraph than $G^*[Z] = H_Z^\#$ of Fig. 9 in all Cases 1–4. □

In what follows, we strengthen Lemma 6.6(iv) and show that any proper 6-cut Z in G^* induces a triangle, i.e., none of Cases 1, 2 or 3 occurs. A proper 6-cut Z in $(G^*; X, Y)$ is called *maximal* if there is no proper 6-cut Z' with $Z \subset Z'$.

LEMMA 6.7. *Let Z be a maximal proper 6-cut in the minimum counterexample $(G^*; X, Y)$, defined by (6.3). If Z satisfies one of Case 1, Case 2, or Case 3 (i.e., $G^*[Z] = H_Z^*$ of Fig. 9(1), (2), and (3), respectively), then the following properties (i)–(v) hold.*

(i) *In Case 1, there is no pair of terminals $t, t' \in X \cup Y$ such that (t, z_2^-), (z_1^+, t),(t', z_3^-), $(z_2^+, t') \in \delta(Z)$. In Case 2, there is no pair of terminals $t, t' \in X \cup Y$ such that $(t, z_2^-), (z_3^+, t)$, $(t', z_3^-), (z_1^+, t') \in \delta(Z)$. In Case 3, there is no pair of terminals $t, t' \in X \cup Y$ such that $(t, z_2^-), (z_3^+, t)$, $(t', z_3^-), (z_2^+, t') \in \delta(Z)$.*

(ii) *In Case 1, assume that there is no terminal t with $(t, z_2^-), (z_1^+, t) \in \delta(Z)$ (without loss of generality by (i)). Then the instance $(G'; X, Y)$ obtained from $(G^*; X, Y)$ by splitting off arcs e_2^- and (z_2^-, z_1^+) at z_2^- is infeasible and irreducible.*

(iii) *In Case 2, assume that there is no terminal t with $(t, z_2^-), (z_3^+, t) \in \delta(Z)$ (without loss of generality by (i)). Then the instance $(G'; X, Y)$ obtained from $(G^*; X, Y)$ by splitting off arcs (z_2^-, z_3^+) and e_3^+ at z_3^+ $(= z_1^-)$ is infeasible and irreducible.*

(iv) *In Case 3, assume that there is no terminal t with $(t, z_2^-), (z_3^+, t) \in \delta(Z)$ or $(t, z_1^-), (z_2^+, t) \in \delta(Z)$ (without loss of generality by (i)). Then the instance $(G'; X, Y)$ obtained from $(G^*; X, Y)$ by splitting off arcs (z_2^-, z_3^+) and e_3^+ at z_3^+ $(= z_1^-)$ is infeasible and irreducible.*

(v) *Let $(G'; X, Y)$ be the instance of (ii) of Case 1 (resp., (iii) of Case 2 and (iv) of Case 3). Then $(G^*; X, Y)$ has an IPR if $(G'; X, Y)$ has an IPR.*

Proof. (i) Assume that there are such terminals t and t' in Cases 1, 2, and 3. By Lemma 3.3(ii), terminals t and t' have degree 2 and $Z \cup \{t, t'\}$ is a 2-cut. In Case 1, G^* would have a cut vertex $z_1^- = z_3^+$ (see Fig. 9(1)), which contradicts Lemma 5.1(i). Then consider Cases 2 and 3. By Lemma 3.3(i), $|\{t, t'\} \cap X| = |\{t, t'\} \cap Y| = 1$ holds, and assume $t = x_1$ and $t' = y_1$ without loss of generality. Furthermore, by Lemma 5.1(ii), we obtain $V - (Z \cup \{t, t'\}) = \{x_2, y_2\}$. Now G^* has $|Z| + 4 = 9$ vertices in Case 2 and $|Z| + 4 = 8$ vertices in Case 3. By inspection, we see that $(G^*; X, Y)$ in Case 2 is feasible or has an IPR and $(G^*; X, Y)$ in Case 3 is feasible, which is a contradiction.

(ii) Obviously $(G'; X, Y)$ is infeasible. We show that $(G'; X, Y)$ has no multiple arcs. If there are such multiple arcs, then they must be $(u, z_1^+), (z_1^+, u)$ for some vertex $u \in V - Z$, since G^* has no multiple arcs by Lemma 3.3(iv). This means that u is

adjacent to both z_2^- and z_1^+ in G^*. By the assumption that there is no terminal t with $(t, z_2^-), (z_1^+, t) \in \delta(Z)$, u is not a terminal. Then $Z \cup \{u\}$ is a proper 6-cut, contradicting the maximality of $|Z|$. Now we apply Lemma 6.1 to $(G'; X, Y)$ and $Z' = Z - \{z_2^-\}$. Clearly, 6-cut Z' satisfies $Z' \cap (X \cup Y) = \emptyset$ and conditions (i) and (iv) of Lemma 6.1. From the irreducibility of G^*, condition (iii) of Lemma 6.1 holds for Z'. By inspection, we see that Z' satisfies condition of (ii) of Lemma 6.1. Therefore, $(G'; X, Y)$ is irreducible by Lemma 6.1.

(iii) This proof is analogous to (ii).

(iv) The assumption that there is no terminal t with $(t, z_2^-), (z_3^+, t) \in \delta(Z)$ or $(t, z_1^-), (z_2^+, t) \in \delta(Z)$ in Case 3 does not lose generality, because if a terminal t is adjacent to both z_2^- and z_3^+, then no terminal is adjacent to both z_3^- and z_2^+ by (i) and two vertices z_2^-, z_3^- cannot be adjacent to another terminal. This assumption ensures that $(G'; X, Y)$ has no multiple arcs. The rest of the proof is analogous to (ii).

(v) It should be noted that if an instance has an IPR, then any triangle (if any) in the instance gives rise to a face in its IPR. Let $Z' = Z - \{z_2^-\}$ in Case 1 and $Z' = Z - \{z_1^-\}$ in Cases 2 and 3. It is easy to see that $(G'; X, Y)$ still contains a triangle in $G'[Z']$ in each of Cases 1, 2, and 3, and the vertices on these triangles are uniquely embedded in an IPR. Based on this, we can observe that if $(G'; X, Y)$ has an IPR, then $(G^*; X, Y)$ has an IPR. □

From this lemma, we can conclude that none of Cases 1, 2, or 3 can happen in $(G^*; X, Y)$ as follows. If situations (ii), (iii), or (iv) occurs, then the instance $(G'; X, Y)$ is irreducible and infeasible, as shown in the lemma. Since the instance $(G'; X, Y)$ is smaller than $(G^*; X, Y)$ and $|V(G')| = |V(G^*)| - 1 \geq 7$, it has an IPR by the assumption on G^* and Lemma 3.6. Then, $(G^*; X, Y)$ also has an IPR by Lemma 6.7(v). This is a contradiction. Therefore, only Case 4 is possible for a maximal proper 6-cut Z (note that Lemma 6.7(iv) no longer holds for Case 4, since $G^*[Z]$ has no triangle after splitting off arcs, say, (z_2^-, z_3^+) and e_3^+ at z_3^+). This implies that any maximal proper 6-cut (and hence any proper 6-cut, which is not necessarily maximal) always induces a triangle.

LEMMA 6.8. *Any proper* 6*-cut in* $(G^*; X, Y)$ *induces a triangle.* □

7. Admissible splitting. In this section, we derive a condition for splitting two arcs at a nonterminal vertex to be admissible (defined in section 4) and then show that $(G^*; X, Y)$ always has an admissible splitting.

LEMMA 7.1. *Let* $(G^*; X, Y)$ *be the minimum counterexample, and let* w *be a nonterminal vertex in* G^**, where* $(s_0, w), (s_1, w), (w, s_2), (w, s_3)$ *are the four arcs incident with* w*. If* s_0 *and* s_2 *are not adjacent, and* s_1 *and* s_3 *are not adjacent, then the instance* $(G_w^*; X, Y)$ *obtained by splitting off* (s_0, w) *and* (w, s_2) *at* w *is connected and irreducible* (*i.e., this splitting is admissible*).

Proof. By Lemma 5.1(i), w is not a cut vertex in G^*, and hence $(G_w^*; X, Y)$ is connected. Assume that $(G_w^*; X, Y)$ has a reducible cut $W \subseteq V - \{w\}$. Since s_0, s_1, s_2, s_3 are distinct by Lemma 3.3(iv), let $S = \{s_0, s_1, s_2, s_3\}$. We see that $W \cap S = \{s_0, s_2\}$ or $W \cap S = \{s_1, s_3\}$ holds (otherwise W would be reducible in $(G^*; X, Y)$). Without loss of generality, assume $W \cap S = \{s_0, s_2\}$ (see Fig. 10). We consider three cases: (a) $|\delta(W; G_w^*)| = 2$ and $W \cap (X \cup Y) = \emptyset$; (b) $|\delta(W; G_w^*)| = 4$, $W \cap (X \cup Y) = \emptyset$, and $|W| \geq 2$; and (c) $|\delta(W; G_w^*)| = 2$, $|W \cap (X \cup Y)| = 1$, and $|W| \geq 2$.

(a) In this case, $W' = W \cup \{w\}$ satisfies $|\delta(W'; G^*)| = |\delta(W; G_w^*)| + 2 = 4$ and $|W'| \geq 2$, implying that W' was reducible in $(G^*; X, Y)$, which is a contradiction.

(b) $W' = W \cup \{w\}$ satisfies $|\delta(W'; G^*)| = |\delta(W; G_w^*)| + 2 = 6$, and $|W'| \geq 3$. Since arcs $(s_1, w), (w, s_3)$ are adjacent to w, W' is a proper 6-cut in G^*, and by Lemma 6.8

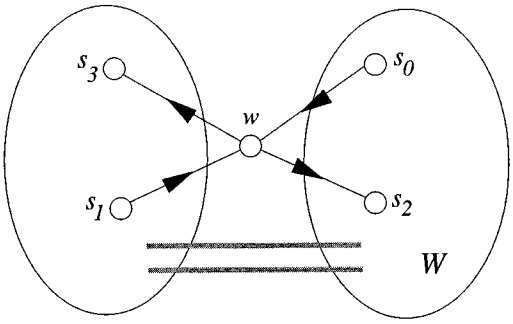


FIG. 10. *Illustration for Lemma* 7.1.

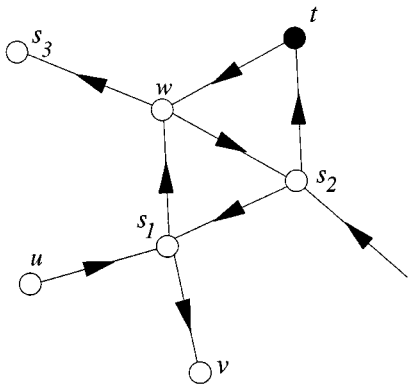


FIG. 11. *Illustration for Lemma* 7.2.

it induces a triangle. However, this contradicts that s_0 and s_2 are not adjacent.

(c) Let $\{t\} = W \cap (X \cup Y)$. We see that $t \neq s_0, s_2$, because otherwise if $t = s_0$ or $t = s_2$, then $W - \{t\}$ is a 4-cut in G^* and $|W - \{t\}| = 1$ must hold by irreducibility of G^*, contradicting that s_0 and s_2 are not adjacent. Then $|W| \geq 3$. This means that $W' = (W - \{t\}) \cup \{w\}$ is a proper 6-cut in G^*, which induces a triangle by Lemma 6.8. However, this again contradicts that s_0 and s_2 are not adjacent. □

LEMMA 7.2. *Let w be a nonterminal vertex adjacent to a terminal t by arc (t, w) in the minimum counterexample $(G^*; X, Y)$, and let $(s_1, w), (w, s_2), (w, s_3)$ be three other arcs incident with w. Then the following property* (i) *or* (ii) *holds.*

(i) *t and s_3 are not adjacent, and s_1 and s_2 are not adjacent (i.e., splitting (t, w) and (w, s_3) at w is admissible by Lemma* 7.1).

(ii) *t and s_2 are not adjacent, and s_1 and s_3 are not adjacent (i.e., splitting (t, w) and (w, s_2) at w is admissible by Lemma* 7.1).

Proof. (a) Consider the case in which t is adjacent to s_2 (i.e., G^* has arc (s_2, t)). Clearly, t cannot be adjacent to s_3. Assume that s_1 and s_2 are adjacent (i.e., G^* has arc (s_2, s_1) by Lemma 6.3(ii)). Let u, v be two other vertices adjacent to s_1, where $(u, s_1), (s_1, v) \in E$ (see Fig. 11). We will show that w (resp., s_2) is not adjacent to u (resp., v). Assume first that w and u are adjacent (i.e., $(w, u) \in E$ by Lemma 6.3(ii), and hence $u = s_3$). If u is a terminal, $W = \{t, u = s_3, w, s_1, s_2\}$ is a 2-cut with $|W \cap (X \cup Y)| = 2$. By Lemma 3.3(i), $|W \cap X| = |W \cap Y| = 1$, and by Lemma 5.1(ii), $|V - W| = 2$, implying $|V - W| + |W| = 7 < n^*$, which is a contradiction. (Recall

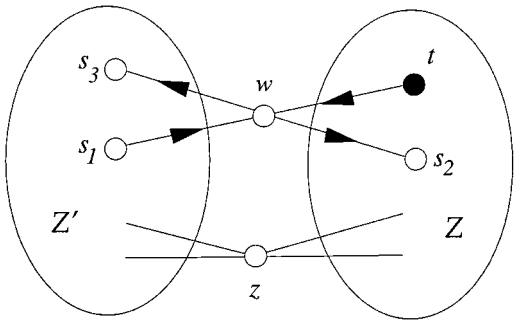


FIG. 12. *Illustration for Lemma* 7.3.

that $n^* \geq 8$ holds by Lemma 3.6, as noted after (4.1).) Then u must be nonterminal. However, in this case $Z = \{u = s_3, w, s_1, s_2\}$ would be a proper 6-cut with four vertices, contradicting Lemma 6.8. Therefore, u is not adjacent to w. Similarly, we see that v is not adjacent to s_2. In other words, splitting off $(u, s_1), (s_1, w)$ at s_1 is admissible by Lemma 7.1 (i.e., the resulting instance $(G^*_{s_1}; X, Y)$ is irreducible by Lemma 7.1). $(G^*_{s_1}; X, Y)$ (containing $n^* - 1 \geq 7$) has an IPR by the assumption on G^* and Lemma 3.6. Since the arcs incident to any terminal lie on the outer face in such IPR, arcs $(s_2, t), (t, w), (w, s_3)$ form part of the boundary of the outer face. Hence, arcs $(u, w), (w, s_2), (s_2, v)$ belong to the boundary of a face in the IPR. This implies that $(G^*; X, Y)$ also has an IPR, which can be obtained by hooking up (u, w) and (s_2, v). This is a contradiction, implying that s_1 and s_2 are not adjacent. Therefore, in this case, we have (i).

(b) If t is adjacent to s_3, we can show that (ii) holds by an analogous argument.

(c) Finally, consider the case in which t is adjacent to neither s_2 or s_3. Assume that s_1 and s_3 are adjacent. We only have to show that s_1 and s_2 are not adjacent. However, if these are adjacent, $Z = \{w, s_1, s_2, s_3\}$ would be a proper 6-cut with four vertices, contradicting Lemma 6.8. □

This lemma says that $(G^*; X, Y)$ always has an admissible splitting at vertex w, which is adjacent to a terminal. We further characterize the digraph obtained by such splitting.

LEMMA 7.3. *Let t be a terminal which is not adjacent to any other terminal, w be a nonterminal vertex adjacent to t by arc (t, w) in the minimum counterexample $(G^*; X, Y)$, and $(s_1, w), (w, s_2), (w, s_3)$ be three other arcs incident with w. Let $G^*_{s_2}$ (resp., $G^*_{s_3}$) denote the instance obtained from G^* by splitting arcs $(t, w), (w, s_2)$ (resp., $(t, w), (w, s_3)$) at w. Then one of these instances is connected and irreducible and has no cut vertex.*

Proof. By Lemma 7.2, one of the instances $G^*_{s_2}$ and $G^*_{s_3}$ is connected and irreducible. Assume without loss of generality that $G^*_{s_2}$ is connected and irreducible, i.e., Lemma 7.2(ii) holds. Then s_1 and s_3 are not adjacent and t and s_2 are not adjacent in G^*. If $G^*_{s_2}$ does not have a cut vertex, then the lemma is shown. Therefore, assume that $G^*_{s_2}$ has a cut vertex z (see Fig. 12).

We first show that w and z are not adjacent in G^* by contradiction. By Lemma 5.1(i), z is not a cut vertex in G^*. Let Z and $Z' = V - \{w, z\} - Z$ be the vertex sets of the two connected components in $G^*_{s_2} - \{z\}$, where $t \in Z$ is assumed. Consider the three possible cases, in which $z = s_1$, $z = s_2$, and $z = s_3$.

First consider the case of $z = s_1$. Then Z' is a 2-cut in G^* and $|Z'| \geq 2$ holds since

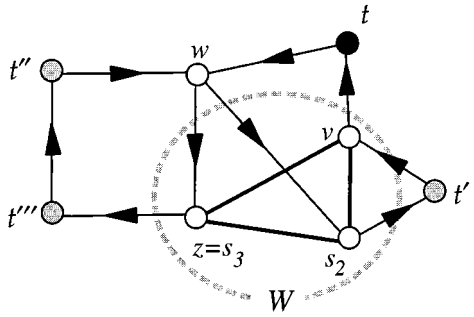


FIG. 13. *The proof of Lemma* 7.3.

s_1 and s_3 are not adjacent. Then $|Z \cap (X \cup Y)| = |Z' \cap (X \cup Y)| = 2$ (otherwise Z' or $V - Z'$ would be a reducible 2-cut in $G^*_{s_2}$). Then, by Lemma 3.3(i), $|Z' \cap X| = |Z' \cap Y| = 1$. If $Z = \{t, s_2\} \subset X \cup Y$, then $|\delta(t; G^*)| = |\delta(s_2; G^*)| = 2$ holds by Lemma 3.3(ii), and in this case we see that $(G^*; X, Y)$ is feasible, which is a contradiction. Then $Z - (X \cup Y) \neq \emptyset$. By this and Lemma 5.1(ii), we have $Z' - (X \cup Y) = \emptyset$ and $|Z'| = 2$. Let t' be the terminal in $Z - \{t\}$. We see that $W = (Z \cup \{w, z\}) - \{t, t'\}$ is a 6-cut in G^* (if W is a 2- or 4-cut, then it would be reducible). By assumption of $n^* \geq 8$, we have $|W| = n^* - 4 \geq 4$, and hence W is a proper 6-cut. However, $|W| = 3$ must hold by Lemma 6.8, which is a contradiction.

Next, consider the case of $z = s_3$. In this case, we can observe $|Z'| = 2$ and $|Z' \cap X| = |Z' \cap Y| = 1$ in a similar manner as in the case of $z = s_1$. Let t' be the terminal in $Z - \{t\}$ and $Z' = \{t'', t'''\}$. We see that $W = (Z \cup \{z\}) - \{t, t'\}$ is a 6-cut in G^* (if W is a 2- or 4-cut, then it would be reducible). By $n^* \geq 8$, $|W| = n^* - 5 \geq 3$ and W is a proper 6-cut. By Lemma 6.8, W induces a triangle. By considering that t and s_2 are not adjacent and G^* has no multiple arc, G^* is given as the instance shown in Fig. 13, where the triangle $W = \{z, s_2, v\}$ has two possible orientations. For any choice of terminals $\{t, t', t'', t'''\}$ from $X \cup Y$ and orientation of the triangle, we can check that the instance is always feasible, which is a contradiction.

Finally, consider the case of $z = s_2$. In this case, we can obtain $|\delta(Z)| = 2$, $|Z| = 2$, and $|Z \cap X| = |Z \cap Y| = 1$ in a similar manner as in the above cases. This implies that t is adjacent to a terminal $\{t'\} = Z - \{t\}$, contradicting the assumption on t of this lemma.

Therefore, w and z are not adjacent in G^*.

Then, $\{s_1, s_3\}$ and $\{t, s_2\}$ are contained in distinct components in $G^*_{s_2} - \{z\}$ since z is a cut vertex in $G^*_{s_2}$ but not in G^*. That is, s_1 and s_2 (resp., t and s_3) are not adjacent, and hence $G^*_{s_3}$ is connected and irreducible by Lemma 7.2.

We show that $G^*_{s_3}$ has no cut vertex. Let $G^*_{s_3}[Z]$ (resp., $G^*_{s_3}[Z']$) denote the subdigraph of $G^*_{s_3}$ induced by Z (resp., Z'). Note that $G^*_{s_3}[Z] = G^*[Z]$ and $G^*_{s_3}[Z'] = G^*[Z']$. Clearly, all vertices in Z (resp., all vertices in Z') are connected in $G^*_{s_3}[Z]$ (resp., $G^*_{s_3}[Z']$) since otherwise G^* would be reducible. This implies that z is no longer a cut vertex in $G^*_{s_3}$. Assume that $G^*_{s_3}$ has another cut vertex z' $(\neq z)$. By a similar argument as above, z' is not equal to any of s_1, s_2, s_3 and $\{s_1, s_2\}$ and $\{t, s_3\}$ are contained in distinct components in $G^*_{s_3} - \{z'\}$. However, this is impossible because if $z' \in Z'$, then t and s_2 are connected in $G^*[Z] (= G^*_{s_3}[Z])$ (without using z'), and otherwise if $z' \in Z$, then s_1 and s_3 are connected in $G^*[Z'] (= G^*_{s_3}[Z'])$ (without using z'). □

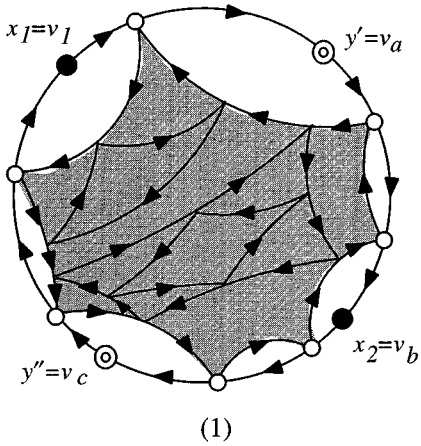


FIG. 14. *Case* 1 *in the proof of Lemma* 8.1 *(the shaded area indicates* $\mathcal{R}'$*; i.e.,* B' *and its interior).*

8. Hooking up arcs in IPR. To complete the proof of Theorem 3.7, this section shows that, given an irreducible infeasible instance G that has an IPR, hooking up any two arcs in G cannot yield G^*. More precisely, hooking up two arcs in G makes G satisfy at least one of the conditions in the following lemma, none of which G^* can satisfy.

LEMMA 8.1. *Let* $(G = (V, E); X, Y)$ *(where* $|V| \geq 7$*) be an irreducible instance which has an IPR and has no cut vertex. For arc* $e = (x_2, u) \in E$ *with* $x_2 \in X$ *and any arc* $e' = (v, v') \in E$*, let* $(G_{e,e'}; X, Y)$ *be the resulting instance obtained by hooking up* e *and* e' *with a new vertex* w*. Then* $G_{e,e'}$ *is connected and one of the following properties* (i)–(v) *holds.*

(i) $G_{e,e'}$ *has a cut vertex.*
(ii) $(G_{e,e'}; X, Y)$ *has a 2-cut* Z *such that* $Z \subseteq V$*,* $|Z \cap X| = |Z \cap Y| = 1$*,* $|Z| \geq 3$*, and* $|(V \cup \{w\}) - Z| \geq 3$*.*
(iii) $(G_{e,e'}; X, Y)$ *is reducible.*
(iv) $(G_{e,e'}; X, Y)$ *has an IPR.*
(v) $(G_{e,e'}; X, Y)$ *is feasible.*

Proof. Assuming that $(G_{e,e'}; X, Y)$ satisfy neither (i) nor (ii), we show that $(G_{e,e'}; X, Y)$ satisfies one of (iii)–(v). Since G has no cut vertex, we only have to consider IPRs as illustrated in Figs. 14, 15, and 16, which correspond respectively to the following three cases.

Case 1. $(G; X, Y)$ has no 2-cut W such that $|W \cap X| = |W \cap Y| = 1$.

Case 2. $(G; X, Y)$ has a 2-cut Z such that $|Z \cap X| = |Z \cap Y| = 1$, $Z - (X \cup Y) \neq \emptyset$, and $(V - Z) - (X \cup Y) \neq \emptyset$, where $x_2 \in Z$.

Case 3. $(G; X, Y)$ has a 2-cut Z such that $|Z \cap X| = |Z \cap Y| = 1$, $Z \subseteq X \cup Y$, or $V - Z \subseteq X \cup Y$, where $x_2 \in Z$.

Let $\mathcal{R}$ be the IPR of $(G; X, Y)$, and let B denote the cycle of the outer face of $\mathcal{R}$, which is a simple cycle since G has no cut vertex. Let $v_1, v_2, \ldots, v_p$ $(p = |V(B)|)$ be the vertices that appear along B clockwise, where $v_1 = x_1, v_a = y', v_b = x_2$, and $v_c = y''$ $(1 < a < b < c)$, and $\{y', y''\} = Y$ are assumed without loss of generality. Let $B(u, v)$ denote the subpath of B from u to v, where $B(u, u)$ means a path of null length. Let $\mathcal{R}'$ denote the planar representation obtained from the IPR of G by eliminating the arcs in $E(B)$ together with $X \cup Y$. We denote components of $\mathcal{R}'$ by

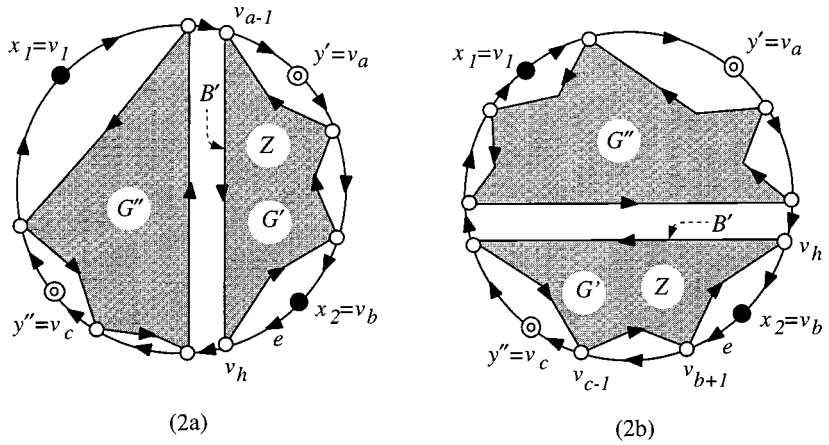


FIG. 15. *Illustration of graph G used in the proof of Lemma* 8.1.

$G', G'', \ldots$ and the directed cycles representing their outer faces by $B', B'', \ldots$. For these cycles, say, B', we denote by $B'(u, v)$ the subpath of B' from u to v. The proof will be given separately for the above three cases.

Case 1 (Fig. 14). In this case, $\mathcal{R}'$ consists of a single component G' by Lemma 3.5. Also, no two terminals in $\mathcal{R}$ are adjacent on B. Note that the directed cycle B' (which may not be simple) visits all vertices in $V(B) - X - Y$ in the order reverse to B (i.e., counterclockwise). Choose $e = (x_2, v_{b+1})$, and partition the arc set $E - e$ into the three subsets

$$E_1 = \{e'' | e'' \text{ is adjacent to } e\},$$
$$E_2 = E(B(v_{b+1}, y'')) \cup E(B'(v_{b+1}, v_{b-1})) - E_1,$$
$$E_3 = E - e - E_1 - E_2 \text{ (see Fig. 15).}$$

It is easy to see that $(G_{e,e'}; X, Y)$ satisfies (iii) (resp., (iv)) for any $e' \in E_1$ (resp., $e' \in E_2$). We then show that (v) holds for all $e' \in E_3$. Since no two terminals in $\mathcal{R}$ are adjacent on B and $\mathcal{R}$ has no cut vertex, G has at least four nonterminal vertices in $V(B')$.

Case 1a. $e' = (v_{a-1}, y') \in E_3$. Then $G_{e,e'}$ has a $y'y''$-path,

$$P_Y = \langle B(y', v_{a+1}), B'(v_{a+1}, v_{a-1}), (v_{a-1}, w), (w, v_{b+1}), B(v_{b+1}, y'') \rangle.$$

Clearly $G_{e,e'} - E(P_Y)$ has an x_1x_2-path, where e and e' are hooked up with vertex w. $P_X = \langle (x_1, v_2), B'(v_2, v_{b-1}), (v_{b-1}, x_2) \rangle$, which implies that $(G_{e,e'}; X, Y)$ is feasible.

Case 1b. $e' = (v_k, v_{k+1}) \in E(B(y', v_{b-1})) \subseteq E_3$. Then $G_{e,e'}$ has a $y'y''$-path

$$P_Y = \langle B(y', v_k), (v_k, w), (w, v_{b+1}), B(v_{b+1}, y'') \rangle.$$

It is also easy to see that x_1 and x_2 are still connected in $G_{e,e'} - E(P_Y)$, implying that $(G_{e,e'}; X, Y)$ is feasible by Lemma 2.1.

Case 1c. $e' \in E_3 - \{(v_{a-1}, y')\} - E(B(y', v_{b-1}))$. In this case, consider the following $y'y''$-chain in G:

$$Q_Y = \langle B(y', v_{b-1}), B'(v_{b+1}, v_{b-1}), B(v_{b+1}, y'') \rangle.$$

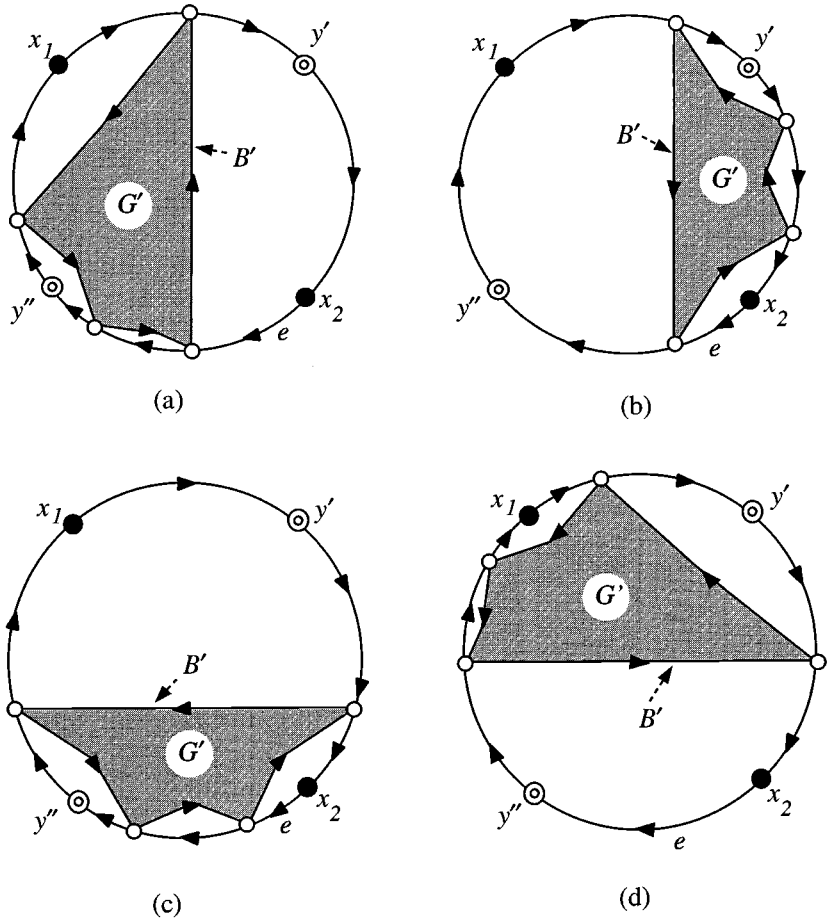


FIG. 16. *Illustration for the proof of Case* 1c *in Lemma* 8.1.

Clearly, $e' \notin E(Q_Y)$ holds, and $G_{e,e'}$ still has $y'y''$-chain Q_Y.

Let $\mathcal{H}$ be the IPR resulting from $\mathcal{R}$ by removing the arcs in $E(B(v_{a-1}, v_{c+1})) \cup E(B'(v_{c+1}, v_{b-1}))$ (see Fig. 16). We now claim that x_1 is reachable in $\mathcal{H}$ from any vertex which is located on the boundary B' or in the area surrounded by B'. By $E(\mathcal{H}) \cap E(Q_Y) = \emptyset$, the claim will mean that, for any $e' = (u', v') \in E_3 - \{(v_{a-1}, y')\} - E(B(y', v_{b-1}))$, $G_{e,e'} - E(Q_Y)$ has a $v'x_1$-path (hence, it has an x_2x_1-path). Then, by Lemma 2.1, this will complete the proof that $(G_{e,e'}; X, Y)$ is feasible. To prove the claim, it is sufficient to show that x_1 is reachable from any vertex on B', since any vertex inside B' is clearly reachable to a vertex on B'.

Partition set $V(B')$ into two subsets $V_1 = V(B'(v_{b-1}, v_{c+1}))$ and $V_2 = V(B') - V_1$. Since two paths $B'(v_{b-1}, v_{c+1})$ and $B(v_{c+1}, v_{a-1})$ remain in $\mathcal{H}$, x_1 is clearly reachable in $\mathcal{H}$ from any vertex $v \in V_1$. We then show that x_1 is reachable from any vertex $v \in V_2$ in $\mathcal{H}$. Let us denote the vertex set $V(B'(v_{c+1}, v_{b-1}))$ $(= V_2 \cup \{v_{c+1}, v_{b-1}\})$ by $\{u_0, u_1, u_2, \ldots, u_q, u_{q+1}\}$, where B' visits these vertices $u_0, \ldots, u_{q+1}$ in this order. Assume that there is a vertex $u_k \in V_2$ which cannot reach any vertex in V_1 and that u_k has the smallest index among such vertices in V_2. We follow the *leftmost* path P^* from u_k in $\mathcal{H}$ until the path returns to u_k (note that P^* must come back to u_k since

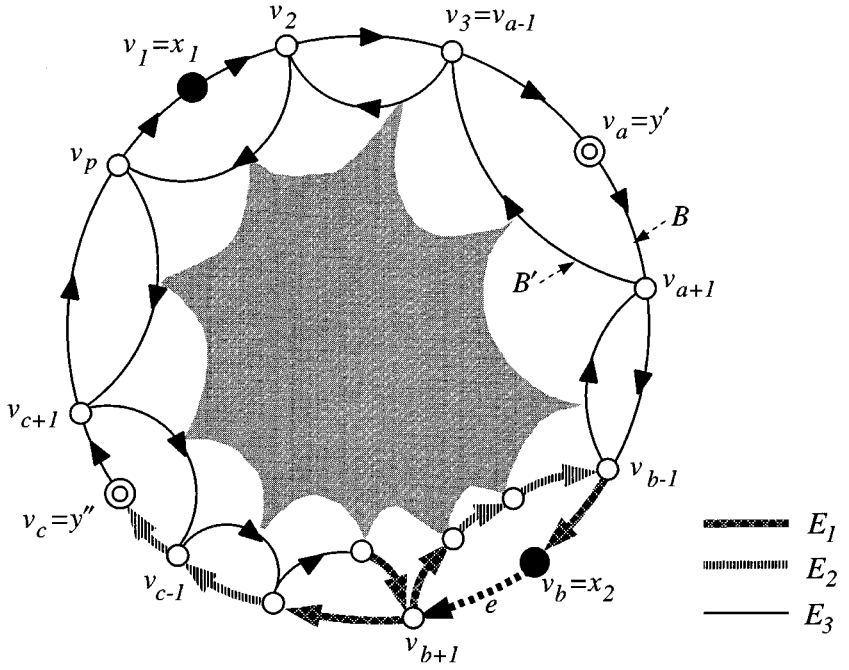


FIG. 17. *Case* 2 *in the proof of Lemma* 8.1.

$indeg(v) = outdeg(v)$ for all $v \in V - V_1$ in $\mathcal{H}$). Clearly, $|V(P^*)| \geq 2$. Let u_h be the first vertex in V_2 that P^* visits. Note that P^* visits no vertex $u_i \in V_2$ with $i > h$. Consider the set V^* of vertices in $V - V(P^*)$ that are adjacent to a vertex of $V(P^*)$ in $\mathcal{H}$. Since the arcs incident to a nonterminal vertex are alternately oriented in and out from the definition of IPR, all vertices in V^* are located in the inside area surrounded by P^*. In other words, V_1 and $V(P^*) \cup V^*$ are disconnected in $\mathcal{H}$. Thus, removal of the four arcs $\{(u_{k-1}, u_k), e_k, (u_h, u_{h+1}), e_h\}$ from G disconnects V_1 and $V(P^*) \cup V^*$, where e_k (resp., e_h) is the arc in $B(x_2, y'')$ such that e_k and (u_{k-1}, u_k) (resp., e_h and (u_h, u_{h+1})) belong to the same face in $\mathcal{R}$. This contradicts that $(G; X, Y)$ has no reducible 4-cut. Therefore, x_1 is reachable from any vertex in $V_1 \cup V_2$.

Case 2 (Fig. 17). In this case, $\mathcal{R}'$ consists of two components G' and G'', where G' (resp., G'') is induced by $V' = Z - (X \cup Y)$ (resp., $V'' = (V - Z) - (X \cup Y)$). There are two cases. Case 2a: $y', x_2 \in Z$ and $y'', x_1 \in V - Z$ (see Fig. 17(2a)), and Case 2b: $x_2, y'' \in Z$ and $x_1, y' \in V - Z$ (see Fig. 17(2b)). In both Cases 2a and 2b, e' must be chosen from $E(G[V - Z])$ to avoid the case (ii) in the lemma statement.

Case 2a(i). $e' \in E(B(x_2, y'')) \cap E(G[V - Z])$. It is easy to see that $(G_{e,e'}; X, Y)$ has an IPR.

Case 2a(ii). $e' \in E(G[V-Z]) - E(B(x_2, y''))$. Let v_h be the vertex in $B(y', y'') \cap Z$ with the largest index, where $b < h < c$ must hold (otherwise V' would be a reducible cut or a cut vertex). Then $G_{e,e'}$ has a $y'y''$-path

$$P_Y = \langle (y', v_{a+1}), B'(v_{a+1}, v_h), B(v_h, y'') \rangle.$$

Furthermore, it is also easy to see that x_1 and x_2 are connected in $G_{e,e'} - E(P_Y)$. Therefore, $(G_{e,e'}; X, Y)$ is feasible by Lemma 2.1.

Case 2b(i). $e' = (v_k, v_{k+1}) \in E(B(y', x_2)) \cap E(G[V - Z])$. Then $G_{e,e'}$ has a $y'y''$-path $P_Y = \langle B(y', v_k), (v_k, w), (w, v_{b+1}), B(v_{b+1}, y'') \rangle$. Obviously, x_1 and x_2 remain

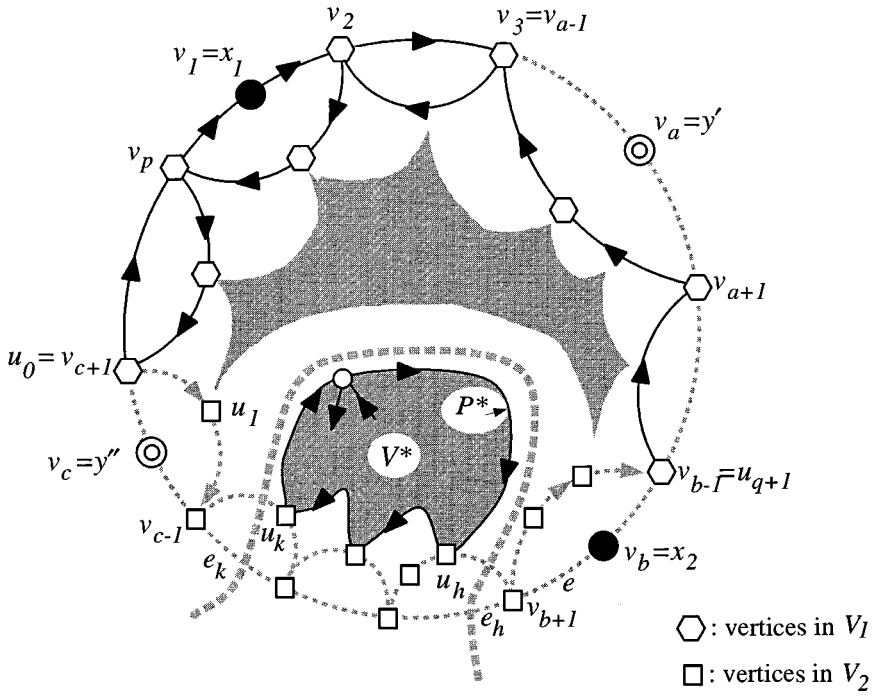


FIG. 18. *Case* 3 *in the proof of Lemma* 8.1.

connected in $G_{e,e'} - E(P_Y)$, implying by Lemma 2.1 that $(G_{e,e'}; X, Y)$ is feasible.

Case 2b(ii). $e' \in E(G[V-Z]) - E(B(y', x_2))$. Let v_h be the vertex in $B(y', y'') \cap Z$ with the smallest index, where $a < h < b$ must hold. Then $G_{e,e'}$ has a $y'y''$-path $P_Y = \langle B(y', v_h), B'(v_h, v_{c-1}), (v_{c-1}, y'')\rangle$. Since it is obvious that x_1 and x_2 are connected in $G_{e,e'} - E(P_Y)$, $(G_{e,e'}; X, Y)$ is feasible by Lemma 2.1.

Case 3 (Fig. 18). In this case, there are two terminals $x^* \in X$ and $y^* \in Y$ which are adjacent on B, and $\mathcal{R}'$ has a single component G'.

Case 3a. $Z = \{y', x_2\}$ (i.e., $(y', x_2) \in E(B)$) (see Fig. 18(3a)). This case can be treated in the same manner as in Case 1, where the corresponding partition of $E - e$ is defined by $E_1 = \{e'' | e''$ is adjacent to $e\}$, $E_2 = \{(v_{a-1}, y')\} \cup E(B'(v_{b+1}, v_{a-1})) \cup E(B(v_{b+1}, y'')) - E_1$, and $E_3 = E - e - E_1 - E_2$ and $y'y''$-chain Q_Y in Case 1c is chosen as $Q_Y = \langle (v_{a-1}, y'), B'(v_{b+1}, v_{a-1}), B(v_{b+1}, y'')\rangle$.

Case 3b. $V - Z = \{y'', x_1\}$ (i.e., $(y'', x_1) \in E(B)$) (see Fig. 18(3b)). This case can be treated in the same manner as in Case 2a.

Case 3c. $V - Z = \{x_1, y'\}$ (i.e., $(x_1, y') \in E(B)$) (see Fig. 18(3c)). This case can be treated in the same manner as in Case 2b.

Case 3d. $Z = \{x_2, y''\}$ (i.e., $(x_2, y'') \in E(B)$) (see Fig. 18(3d)). This case can be treated in the same manner as in Case 1, where the corresponding partition of $E - e$ is defined by $E_1 = \{e'' | e''$ is adjacent to $e\}$, $E_2 = E(B'(v_{c+1}, v_{b-1}))$, and $E_3 = E - e - E_1 - E_2$ and $y'y''$-chain Q_Y in Case 1c is chosen as $Q_Y = \langle B(y', v_{b-1}), B'(v_{c+1}, v_{b-1}), B(y'', v_{c+1})\rangle$.

Now we are ready to prove Theorem 3.7 by deriving a contradiction from the assumption that a minimum counterexample $(G^*; X, Y)$ exists. G^* must have a nonterminal vertex adjacent to a terminal (otherwise, G^* consists of only terminals, contradicting $n^* \geq 8$). We can assume without loss of generality that $(G^*; X, Y)$ has a

terminal x_2 which is not adjacent to any other terminal (because if any terminal is adjacent to some other terminal, then $|\delta(V-(X\cup Y))| \leq 4$, and hence $|V-(X\cup Y)| \leq 1$ would hold by irreducibility of G^*). Then by Lemma 7.3, two arcs (x_2, w) and (w, s_2) in $(G^*; X, Y)$ can be split off at w so that the resulting instance $(G^*_w; X, Y)$ is still connected and irreducible, and has no cut vertex, where (s_1, w) and (w, s_3) are the other arcs incident to w. In other words, G^* can be obtained from G^*_w by hooking up two arcs $e = (x_2, s_2)$ and $e' = (s_1, s_3)$ after introducing w. We then apply Lemma 8.1 to $G = G^*_w$ and $G_{e,e'} = G^*$. By Lemma 5.1, neither (i) nor (ii) of Lemma 8.1 holds for G^*. Furthermore, none of the remaining (iii)–(v) of Lemma 8.1 is possible by the definition of G^*. This is a contradiction and proves the next lemma.

LEMMA 8.2. *Let $(G; X, Y)$ be an infeasible irreducible instance that satisfies $|V| \neq 6$. Then $(G; X, Y)$ has an IPR.* □

Finally, this proves Theorem 3.7, since, by Lemma 3.2, this is a stronger statement than Theorem 3.7.

9. Complexity results. Based on Theorem 3.7 (or Lemma 8.2 to be more precise), we can test if a given instance $(G; X, Y)$ is feasible or not in polynomial time.

LEMMA 9.1. *Given an instance $(G; X, Y)$, one of its irreducible instances $(G'; X, Y)$ can be found in $O(m + n \log n)$ time, where n and m denote the numbers of vertices and arcs in G, respectively.* □

Before describing the algorithm for computing an irreducible instance, let us review a *cactus representation* [1], a compact representation of all minimum cuts in an undirected graph. A connected undirected graph is called a *cactus* if, for each edge, there is exactly one simple cycle that contains it, where the cycle may be of length 2. Then, in a cactus, two cycles (if any) have at most one common vertex, which is a cut vertex. A vertex with degree 2 in a cactus is called a *leaf vertex*. Given an undirected graph $G = (V, E)$, we map it to a cactus $\Gamma = (W, F)$ by a mapping $\varphi : V \to W$, where φ may not be an onto-mapping. The size of a minimum cut in G (resp., in Γ) is defined by $\lambda(G) = \min\{|\delta(Z; G)| \mid \emptyset \neq Z \subset V\}$ (resp., $\lambda(\Gamma) = \min\{|\delta(S; \Gamma)| \mid \emptyset \neq S \subset W\}$), where $\delta(Z; G)$ denotes the set of edges between Z and $V - Z$ in G (similarly for $\delta(S; \Gamma)$). Clearly, in a cactus $\Gamma = (W, F)$ with $|W| \geq 2$, $\lambda(\Gamma) = 2$ holds.

Let $\mathcal{C}(G) = \{Z \mid \emptyset \neq Z \subset V, |\delta(Z; G)| = \lambda(G)\}$ and $\mathcal{C}(\Gamma) = \{S \mid \emptyset \neq S \subset W, |\delta(S; \Gamma)| = \lambda(\Gamma)\}$ denote the sets of all minimum cuts of G and Γ, respectively. Note that S belongs to $\mathcal{C}(\Gamma)$ if and only if two arcs in $\delta(S; \Gamma)$ belong to the same cycle. In the following description, we use the term "vertex" to denote an element in V and the term "node" to denote an element in W. There may be a node $x \in W$ with $\varphi^{-1}(x) = \emptyset$, which is called an empty node. Define

$$\varphi(Z) \equiv \{\varphi(v) \in W \mid v \in Z\} \quad \text{for } Z \subseteq V \text{ and}$$
$$\varphi^{-1}(S) \equiv \{v \in V \mid \varphi(v) \in S\} \quad \text{for } S \subseteq W.$$

A pair (Γ, φ) of a cactus and a mapping φ is called a cactus representation for $\mathcal{C}(G)$ if it satisfies (i) and (ii) below.

(i) For any cut $Z \in \mathcal{C}(G)$, there exists a cut $S \in \mathcal{C}(\Gamma)$ such that $Z = \varphi^{-1}(S)$ and $V - Z = \varphi^{-1}(W - S)$.

(ii) Conversely, for any 2-cut $S \in \mathcal{C}(\Gamma)$, $Z = \varphi^{-1}(S)$ satisfies $Z \in \mathcal{C}(G)$.

It is known [1] that $G = (V, E)$ always has such a cactus representation $(\Gamma = (W, F), \varphi)$ with $|W| = O(|F|) = O(|V|)$, which can be constructed in $O(|E| + \lambda(G)^2|V| \log |V|)$ time [7]. We say that a cut $Z \in \mathcal{C}(G)$ and a cut $S \in \mathcal{C}(\Gamma)$ correspond to each other if $Z = \varphi^{-1}(S)$ and $V - Z = \varphi^{-1}(W - S)$. Note that if Γ has an

empty node, a minimum cut in $\mathcal{C}(G)$ may correspond to more than one minimum cut in $\mathcal{C}(\Gamma)$, while any minimum cut in $\mathcal{C}(\Gamma)$ always corresponds to exactly one minimum cut in $\mathcal{C}(G)$. Obviously, any leaf node $w \in W$ corresponds to a minimum cut in $\mathcal{C}(G)$, and there are at least two leaf nodes in Γ.

LEMMA 9.2. *For an undirected graph $G = (V, E)$ and a designated vertex $t^* \in V$, let $\mathcal{Z} = \{Z_1, Z_2, \ldots, Z_q\}$ be the set of all cuts Z_i such that*

(i) *$Z_i \in \mathcal{C}(G)$ and $Z_i \subseteq V - \{t^*\}$,*

(ii) *$|Z_i|$ is maximal subject to* (i) *(i.e., no cut $Z' \in \mathcal{C}(G)$ with $Z_i \subset Z' \subseteq V - \{t^*\}$). Then any two cuts $Z_i, Z_j \in \mathcal{Z}$ are mutually disjoint, and the set $\mathcal{Z}$ can be computed in $O(|E| + \lambda(G)^2|V| \log |V|)$ time.*

Proof. Consider a cactus representation $(\Gamma = (W, F), \varphi)$ for $\mathcal{C}(G)$. Let $w^* = \varphi(t^*) \in W$, and let Γ have p cycles passing through w^*. In other words, removal of w^* from Γ creates p connected components with node sets W_i, $i = 1, 2, \ldots, p$. Let $Z_i = \varphi^{-1}(W_i)$, $i = 1, \ldots, p$. Since each W_i is a 2-cut in Γ, we have $W_i \in \mathcal{C}(\Gamma)$ and hence $Z_i \in \mathcal{C}(G)$ by definition of a cactus representation. Hence, each Z_i satisfies condition (i). If there is a cut $Z' \in \mathcal{C}(G)$ such that $Z_i \subset Z' \subseteq V - \{t^*\}$, then there is a cut $W' \in \mathcal{C}(\Gamma)$ such that $Z' = \varphi^{-1}(W')$, $W_i \subset W'$, and $w^* \in W - W'$. However, Γ cannot have any such 2-cut W' separating w^* and W_i by the definition of W'. Therefore, each Z_i satisfies condition (ii). Obviously, $Z_i, \ldots, Z_p$ are mutually disjoint by disjointness of $W_1, \ldots, W_p$. The stated time complexity follows from the fact that a cactus representation $(\Gamma = (W, F), \varphi)$ with $|W| + |F| = O(|V|)$ can be obtained in $O(|E| + \lambda(G)^2|V| \log |V|)$ time [7], and computing connected components in $\Gamma - w^*$ can be done in $O(|W| + |F|) = O(|V|)$ time. □

Proof of Lemma 9.1. Given an instance $(G; X, Y)$, where $n = |V|$ and $m = |E|$, the following algorithm applies all reductions of type (1), (2), and (3), defined in the beginning of section 3.

1. Type (1) reductions (i.e., 2-cuts Z such that $|Z| \geq 1$ and $Z \cap (X \cup Y) = \emptyset$): We contract four terminals x_1, x_2, y_1, y_2 into a single vertex t^* and ignore arc orientation in G. Let $\overline{G}$ denote the resulting undirected graph. Clearly, $\lambda(\overline{G}) \geq 2$, since G is connected and Eulerian. It is easy to see that a cut $Z \subseteq V - \{x_1, x_2, y_1, y_2\}$ is 2-cut in G if and only if $\lambda(\overline{G}) = 2$, $Z \in \mathcal{C}(\overline{G})$, and $Z \subseteq V(\overline{G}) - \{t^*\}$. We can check if $\lambda(\overline{G}) \geq 2$ in $O(m + n \log n)$ time [7]. If $\lambda(\overline{G}) > 2$, then there is no cut of type (1), and we go to 2. If $\lambda(\overline{G}) = 2$, then by Lemma 9.2 the set $\{Z_1, \ldots, Z_p\}$ of these cuts Z with maximal $|Z|$ is uniquely determined and obtained in $O(m + n \log n)$ time. Apply reduction (1) to all cuts Z_i in $(G; X, Y)$. This can be done in $O(m + n)$ time (since $Z_i \cap Z_j = \emptyset$ for $1 \leq i < j \leq p$ if $p \geq 2$). Go to 2 after letting $(G; X, Y)$ be the resulting instance.
2. Type (2) reductions (i.e., 2-cuts Z such that $|Z| \geq 2$ and $|Z \cap (X \cup Y)| = 1$): For each terminal $t \in X \cup Y$, let $\overline{G}_t$ denote the undirected graph obtained from G by contracting the other three terminals $X \cup Y - \{t\}$ into a single vertex $\bar{t}$ and ignoring arc orientations. We easily see that if G has a 2-cut Z with $Z \cap (X \cup Y) = \{t\}$ and $|Z| \geq 2$, then $\lambda(\overline{G}_t) = 2$, $Z \in \mathcal{C}(\overline{G}_t)$, and $Z \subseteq V(\overline{G}) - \{\bar{t}\}$ hold. Then such Z contains t (otherwise, Z would be a cut of type (1), which has been eliminated in the above 1) and is unique if it is maximal (since at most one cut can contain t). Furthermore, such Z can be obtained in $O(m + n \log n)$ time; see Lemma 9.2. We apply reduction (2) to the cut Z in $(G; X, Y)$ in $O(m + n)$ time. This procedure for all four terminals can be done in $O(m + n \log n)$ time. Go to 3 after letting $(G; X, Y)$ be the resulting instance.

3. Type (3) reductions (i.e., 4-cuts Z such that $G[Z]$ is connected, $|Z| \geq 2$, and $Z \cap (X \cup Y) = \emptyset$): Since the current $(G; X, Y)$ has no cut of type (1), any 4-cut $Z \subseteq V - (X \cup Y)$ induces a connected subdigraph $G[Z]$. Let $\overline{G}$ be the undirected graph obtained from $(G; X, Y)$ by contracting four terminals x_1, x_2, y_1, y_2 into a single vertex t^* and ignoring arc orientations. Clearly, $\lambda(\overline{G}) \geq 4$ (otherwise, $(G; X, Y)$ would have a reducible 2-cut). We easily see that a cut $Z \subseteq V - (X \cup Y)$ is 4-cut in G if and only if $\lambda(\overline{G}) = 4$, $Z \in \mathcal{C}(\overline{G})$, and $Z \subseteq V(\overline{G}) - \{t^*\}$. We can check if $\lambda(\overline{G}) \geq 4$ in $O(m + n \log n)$ time. If $\lambda(\overline{G}) > 4$, then there is no cut of type (3) and the current instance $(G; X, Y)$ is irreducible. If $\lambda(\overline{G}) = 4$, then by Lemma 9.2 the set $\{Z_1, \ldots, Z_p\}$ of these cuts Z with maximal $|Z|$ is uniquely determined and is obtained in $O(m + n \log n)$ time. Apply reduction (3) to all these cuts Z_i in $(G; X, Y)$ to obtain an irreducible instance. This can be done in $O(m + n)$ time. □

Given an irreducible instance $(G'; X, Y)$, we can check if it is feasible or not in linear time as follows. If G' has less than 7 vertices, its feasibility can be easily checked in $O(1)$ time (since any irreducible infeasible digraph G' with $|V| < 7$ has $O(1)$ arcs). Otherwise, test if the resulting irreducible instance $(G'; X, Y)$ has an IPR, which can be done in $O(m + n)$ time by using a fast planar drawing algorithm [8]. If it has an IPR, then it is infeasible; otherwise it is feasible. Therefore, we have established the next theorem.

THEOREM 9.3. *Given an instance $(G; X, Y)$, where n and m are the numbers of vertices and arcs, respectively, testing if it is feasible or not can be done in $O(m + n \log n)$ time.* □

We now show that, if a given instance $(G; X, Y)$ is feasible, a solution (i.e., a pair of arc-disjoint $x'x''$- and $y'y''$-paths in G, where $\{x', x''\} = X$ and $\{y', y''\} = Y$) can be found in $O(m(m + n \log n))$ time.

Let $(G = (V, E); X, Y)$ be an irreducible feasible instance. If V consists of only four terminals, then a solution is easily found in $O(1)$ time. Otherwise, one of the following four cases A–D occurs, and we can find a pair of arcs such that the instance remains feasible after splitting them off.

A. There is a nonterminal vertex v with $deg(v) \leq 6$ or a terminal v with $deg(v) = 4$ in instance $(G; X, Y)$: Choose such a vertex v, and find two arcs $(v', v) \in \delta^-(v)$ and $(v, v'') \in \delta^+(v)$ such that the instance obtained by splitting (v', v) and (v, v'') at v remains feasible. Note that such a pair of arcs exists (since the instance is feasible), and by Theorem 9.3 it is found in $O(m + n \log n)$ time by checking feasibility among all (at most 9) possibilities. Split off such (v', v) and (v, v'') at v, and recompute an irreducible instance $(G'; X, Y)$ from the resulting instance in $O(m + n \log n)$ time (Lemma 9.1).

B. $deg(v) \geq 8$ for all $v \in V - (X \cup Y)$, $deg(t) \neq 4$ for all $t \in X \cup Y$, and there is a 6-cut $Z \subseteq V - (X \cup Y)$: Then choose a 6-cut Z with minimal $|Z|$ among such 6-cuts, and let v be a nonterminal vertex in Z. Note that any nonempty cut

$$Z^* \subset Z \text{ satisfies } |\delta(Z^*; G)| \geq 8 \text{ from the assumption on } Z.$$

Since $|\delta^-(v)| = |\delta^+(v)| \geq 4$ by $deg(v) \geq 8$ and $|\delta^-(Z)| = |\delta^+(Z)| = 3$, there are arcs $(v', v) \in \delta^-(v) - \delta^-(Z)$ and $(v, v'') \in \delta^+(v) - \delta^+(Z)$, where $v', v'' \in Z$ (possibly $v' = v''$). Let $(G'; X, Y)$ be the instance obtained from $(G; X, Y)$ by splitting off these arcs (v', v) and (v, v'') at v (in the case of $v' = v''$, splitting simply means removal of those two arcs). We show that $(G'; X, Y)$

remains irreducible (hence feasible, because any irreducible instance having a nonterminal vertex v with $deg(v) \geq 6$ is feasible by Lemma 3.3(iii)). Assume that $(G'; X, Y)$ has a reducible cut Z'. Since Z' was not reducible in $(G; X, Y)$, Z' must separate $\{v\}$ and $\{v', v''\}$. Since $Z' \subset Z$ would imply $|\delta(Z'; G)| \geq 8$ from the above and $|\delta(Z'; G')| \geq 6$ (hence, such Z' is not reducible in G'), Z' must intersect Z (and hence Z' and Z cross each other because $(V - (Z' \cup Z))$ contains a terminal). From the above, we have $|\delta(Z \cap Z'; G)| \geq 8$ and

$$|\delta(Z \cap Z'; G')| \geq 6 \ (= |\delta(Z; G')|).$$

Also, we obtain

$$|\delta(Z \cup Z'; G')| \geq |\delta(Z'; G')| + 2$$

(otherwise $|\delta(Z \cup Z'; G')| \leq |\delta(Z'; G')|$ implies that $Z \cup Z'$ is a reducible cut in G). However, these two inequalities contradict (2.1) (i.e., $|\delta(Z; G')| + |\delta(Z'; G')| \geq |\delta(Z \cap Z'; G')| + |\delta(Z \cup Z'; G')|$). This shows that $(G'; X, Y)$ is irreducible, and hence feasible.

A minimal 6-cut Z in the above can be found in $O(m + n \log n)$ time as follows. Since such Z never intersects $X \cup Y$, we contract the four terminals into a single vertex t^* and ignore the arc orientation. Let $\overline{G_{t^*}}$ be the resulting undirected graph. Clearly, $\lambda(\overline{G_{t^*}}) = 6$ by the irreducibility of G and the assumption of case B. Find a cactus representation (Γ, φ) for $\mathcal{C}(\overline{G_{t^*}})$ in $O(m + n \log n)$ time [7]. Recall that Γ has at least two leaf nodes, and one of them, say, z, satisfies $t^* \notin \varphi^{-1}(z)$. By definition of a cactus representation, $Z = \varphi^{-1}(z)$ is a minimal 6-cut in G.

C. $deg(v) \geq 8$ for all $v \in V - (X \cup Y)$, $deg(t) \neq 4$ for all $t \in X \cup Y$, and $|\delta(Z^*)| \geq 8$ for all $Z^* \subseteq V - (X \cup Y)$, but there is a 4-cut Z with $Z \cap (X \cup Y) = \{t\}$ for some terminal t: Then take a minimal Z among them. Since $deg(t) \neq 4$, we see that $Z - \{t\} \neq \emptyset$ and $deg(t) \geq 6$ (if $deg(t) = 2$, then $Z - \{t\}$ is a 6-cut with $Z - \{t\} \subseteq V - (X \cup Y)$, contradicting the assumption of case C). Since $|\delta^-(t)| = |\delta^+(t)| \geq 3$ by $deg(t) \geq 6$ and $|\delta^-(Z)| = |\delta^+(Z)| = 2$, there are arcs $(v', t) \in \delta^-(\{t\}) - \delta^-(Z)$ and $(t, v'') \in \delta^+(\{t\}) - \delta^+(Z)$, where $v', v'' \in Z$ (possibly $v' = v''$). Let $(G'; X, Y)$ be the instance obtained from $(G; X, Y)$ by splitting off these arcs (v', t) and (t, v'') at t (in the case of $v' = v''$, splitting means removal of those two arcs). We show that $(G'; X, Y)$ remains irreducible (hence feasible, because any irreducible instance having a terminal vertex t with $deg(t) \geq 4$ is feasible by Lemma 3.3(ii)). Assume that $(G'; X, Y)$ has a reducible cut Z'. Since Z' is not reducible in $(G; X, Y)$, Z' must separate $\{t\}$ and $\{v', v''\}$. Since $Z' \subset Z$ implies that $|\delta(Z')| \geq 6$ (if $t \in Z'$) by the minimality of $|Z|$ and $|\delta(Z')| \geq 8$ (if $t \notin Z'$) by the assumption of case C (hence, such Z' is not reducible in G'), then Z' intersects Z (and hence Z' and Z cross each other since $(V - (Z' \cup Z))$ contains a terminal). We see that Z' is not a cut of type (1) because otherwise Z' would be a reducible 4-cut in $(G; X, Y)$. If Z' is a cut of type (3) in $(G'; X, Y)$, then Z' is a 6-cut $Z' \subseteq V - (X \cup Y)$ in $(G; X, Y)$, contradicting the assumption of case C. Therefore, Z' must be a reducible cut of type (2) in $(G'; X, Y)$. We first consider the case of $t \in Z \cap Z'$. We have $|\delta(Z \cap Z'; G)| \geq 6$ (by the minimality of $|Z|$) and $|\delta(Z \cap Z'; G')| \geq 4 (= |\delta(Z; G')|)$. Since $Z - Z'$ contains no terminal, we obtain $|\delta(Z \cup Z'; G')| \geq |\delta(Z'; G')| + 2$ (otherwise $|\delta(Z \cup Z'; G)| = |\delta(Z \cup Z'; G')| \leq |\delta(Z'; G')|$ implies that $Z \cup Z'$ is a reducible

cut in G). However, these two inequalities contradict (2.1), as in case B. Then assume $t \in Z - Z'$, implying that there is another terminal t' in $Z' - Z$. Clearly, $|\delta(Z - Z'; G)| \geq 6$ $(= |\delta(Z; G)| + 2)$ (from minimality of Z). From $|\delta(Z'; G')| = 2$, we have $|\delta(Z'; G)| = 4$ and $deg(t') \geq 6$ (otherwise, if $deg(t') = 2$, then $Z' - \{t'\}$ would be a 6-cut, contradicting the assumption of case C). From this and the irreducibility of $(G; X, Y)$, $|\delta(Z' - Z; G)| \geq 4$ $(= |\delta(Z'; G)|)$ holds. These inequalities contradict (2.2). Consequently, $(G'; X, Y)$ is irreducible and hence is feasible.

The above minimal 4-cut Z can be found in $O(m + n \log n)$ time as follows. Since such Z always separates $\{t\}$ and $(X \cup Y) - \{t\}$, contract the three other terminals of $(X \cup Y) - \{t\}$ into a single vertex $\bar{t}$ and ignore the arc orientation. Let $\overline{G_{\bar{t}}}$ be the resulting undirected graph. Clearly, $\lambda(\overline{G_{\bar{t}}}) = 4$ (since case C does not occur for this t if $\lambda(\overline{G_{\bar{t}}}) \geq 6$). Find a cactus representation (Γ, φ) for $\mathcal{C}(\overline{G_{\bar{t}}})$ in $O(m + n \log n)$ time [7]. By definition of a cactus representation, Γ has a leaf node z with $t \in \varphi^{-1}(z)$ and $Z = \varphi^{-1}(z)$ is a desired minimal 4-cut in G.

D. $deg(v) \geq 8$ for all $v \in V - (X \cup Y)$, $deg(t) \neq 4$ for all $t \in X \cup Y$, and $|\delta(Z^*)| \geq 8$ for all $Z^* \subseteq V - (X \cup Y)$ and $|\delta(Z)| \geq 6$ for all Z with $Z \cap (X \cup Y) = \{t\}$ and $t \in X \cup Y$: Then choose an arbitrary nonterminal vertex v and two arcs (v', v) and (v, v''). It is easy to see that the instance $(G'; X, Y)$ obtained from $(G; X, Y)$ by splitting off these arcs (v', v) and (v, v'') at v remains irreducible (hence feasible, because any irreducible instance having a nonterminal vertex v with $deg(v) \geq 6$ is feasible by Lemma 3.3(iii)).

Recall that none of cases A–D can be applied to an instance only when it has four terminals with degree 2 but no nonterminal vertex. Given an irreducible feasible instance, we continue to split off a pair of arcs to obtain smaller feasible instances by following the above cases A–D until an instance consisting of four terminals with degree 2 is obtained, in which we can easily find a solution. The entire running time of this procedure is $O(m(m + n \log n))$, since the number of arcs decreases at least by 2 after splitting off a pair of arcs. It is easy to see that a solution of the original instance $(G; X, Y)$ can be recovered in the same time complexity from the sequence of such splittings. This establishes the next theorem.

THEOREM 9.4. *Given a feasible instance $(G; X, Y)$, where n and m are the numbers of vertices and edges, respectively, a solution of $(G; X, Y)$ can be computed in $O(m(m + n \log n))$ time.* □

10. Discussion. For the arc-disjoint path problems

$$(G; X_i = \{s_i, t_i\}, i = 1, 2, \ldots, k)$$

associated with Eulerian digraphs, different problem settings are conceivable depending upon the restrictions on G and the directions of the required paths: (i) either $G + H$ is Eulerian, where H is the demand digraph, or G itself is Eulerian, and (ii) either $s_i t_i$-paths are required for all i, or one of the $s_i t_i$- and $t_i s_i$-paths is required for each i. The result in [9] shows that ($G + H$ Eulerian, $s_i t_i$-path, $k = 3$) can be solved in polynomial time, while our result here shows that (G Eulerian, one of the $s_i t_i$- and $t_i s_i$-paths, $k = 2$) can also be solved in polynomial time. By generalizing the proof in [4, 9], it is possible to prove that all types become NP-hard if k is considered as a part of input. Therefore, an interesting theoretical challenge, for each problem type, will be to find out the maximum constant k that permits a polynomial time algorithm, or to show that any constant k permits a polynomial time

algorithm.

Acknowledgment. We wish to thank the anonymous referee for his helpful comments.

REFERENCES

[1] E. A. Dinits, A. V. Karzanov, and M. V. Lomonosov, *On the structure of a family of minimal weighted cuts in a graph*, in Studies in Discrete Optimization, A.A. Fridman, ed., Nauka, Moscow, 1976, pp. 290–306 (in Russian).

[2] E. A. Dinits and A. V. Karzanov, *On existence of two edge-disjoint chains in multi-graph connecting given pairs of its vertices*, Graph Theory Newsletters, 8 (1979), pp. 2–3.

[3] E. A. Dinits and A. V. Karzanov, *On two integer flows of value* 1 *in a network*, in Combinatorial Methods for Network Flow Problems, A.V. Karzanov, ed., Institute for System Studies, Moscow, 1979, pp. 127–137 (in Russian).

[4] S. Fortune, J. E. Hopcroft, and J. Wyllie, *The directed subgraph homeomorphism problem*, Theoret. Comput. Sci., 10 (1980), pp. 111–121.

[5] A. Frank, *On connectivity properties of Eulerian digraphs*, in Graph Theory, in Memory of G. A. Dirac, Ann. Discrete Math. 41, North–Holland, Amsterdam, 1988, pp. 179–194.

[6] A. Frank, *Graph connectivity and network flows*, in Handbook of Combinatorics I, R. L. Graham, M. Grötschel, and L. Lovász, eds., North–Holland, Amsterdam, 1995, pp. 111–177.

[7] H. N. Gabow, *Applications of a poset representation to edge connectivity and graph rigidity*, in Proc. 32nd IEEE Symp. Found. Comp. Sci., San Juan, Puerto Rico, 1991, IEEE Computer Society Press, Los Alamitos, CA, pp. 812–821.

[8] J. E. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, J. Assoc. Comput. Math., 21 (1974), pp. 549–568.

[9] T. Ibaraki and S. Poljak, *Weak three-linking in Eulerian digraphs*, SIAM J. Discrete Math., 4 (1991), pp. 84–98.

[10] N. Robertson and P. D. Seymour, *Graph minors* XIII: *The disjoint path problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.

[11] P. D. Seymour, *Disjoint paths in graphs*, Discrete Math., 29 (1980), pp. 239–309.

[12] C. Thomassen, 2*-linked graphs*, European J. Combin., 1 (1980), pp. 371–378.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 590–602, November 1998

005

MULTIPLE CAPACITY VEHICLE ROUTING ON PATHS*

D. J. GUAN† AND XUDING ZHU†

Abstract. Consider the problem of transporting a set of objects between the vertices of a simple graph by a vehicle that traverses the edges of the graph. The problem of finding a shortest tour for the vehicle to transport all objects from their initial vertices to their destination vertices is called the *vehicle routing problem*. The problem is *multiple capacity* if the vehicle can handle more than one objects at a time. The problem is *preemptive* if objects can be unloaded at the intermediate vertices. In this paper, we present an $O(kn + n^2)$ time algorithm for multiple capacity preemptive vehicle routing problem on paths, where k is the number of objects to be moved and n is the number of vertices in the path.

Key words. vehicle routing, motion planning, graph algorithms

AMS subject classifications. 05C85, 05C90, 68Q25

PII. S0895480197319951

1. Introduction.

Consider an edge weighted graph with objects located at some vertices. Associated with each object is a destination vertex to which that object is to be moved by a vehicle that traverses the edges of the graph. A fundamental problem in motion planning is to determine a tour of minimum cost for the vehicle to transport all objects from their initial vertices to their destination vertices. The determination of a minimum cost tour for the problem is called the *vehicle routing problem*, and we are interested in the computational complexity of the vehicle routing problem.

One factor that affects the complexity of the vehicle routing problem is whether or not we allow drops in the process of transportation. A *drop* is an unloading of an object at a vertex that is not its destination. If an object is dropped, its movement is not immediately completed, and the object will be picked up and transported farther at some later time in the transportation. Based on whether or not we allow drops in the transportation, we have two versions of the vehicle routing problem. We shall use the term *nonpreemptive* to denote the version in which no objects can be dropped at intermediate vertices and the term *preemptive* to denote the version in which objects can be dropped at intermediate vertices.

Another factor that makes a difference on the complexity of the vehicle routing problem is whether the capacity of the vehicle is 1 or greater than 1. Suppose the vehicle can transport c objects at a time. We refer to the problem as the *unit capacity vehicle routing problem* if $c = 1$ and refer to it as the *multiple capacity vehicle routing problem* if $c \geq 2$. Combined with the preemptive and nonpreemptive versions, we have four kinds of vehicle routing problems: unit capacity preemptive vehicle routing; unit capacity nonpreemptive vehicle routing; multiple capacity preemptive vehicle routing; and multiple capacity nonpreemptive vehicle routing.

For general graphs, all four problems are NP-hard [6]. However, for practical applications such as in elevators and those applications that arise in robotics, it suffices to consider more restricted classes of graphs. Indeed, for all of the recent research

*Received by the editors April 21, 1997; accepted for publication May 6, 1998; published electronically September 1, 1998. This research was supported in part by National Science Council of the Republic of China grants NSC87-2115-M-110-003 and NSC87-2115-M-110-004.
http://www.siam.org/journals/sidma/11-4/31995.html
†Department of Applied Mathematics, National Sun Yat Sen University, Kaoshiung 80424, Taiwan, Republic of China (guan@math.nsysu.edu.tw, zhu@math.nsysu.edu.tw).

in this area, focus has been on solving the vehicle routing problems on paths, cycles, and trees [1, 3, 4, 5, 2, 7].

The unit capacity (preemptive and nonpreemptive) vehicle routing problems on paths and cycles are shown to be polynomial time decidable by Atallah and Kosaraju [1] and Frederickson [3]. For unit capacity vehicle routing problems on trees, Frederickson and Guan [4] proved that the preemptive version is polynomial time decidable, while the nonpreemptive version is NP-complete.

For multiple capacity nonpreemptive vehicle routing problems, Guan proved that the problem is NP-complete even if the underlying graph is a path [2]. This, of course, implies that the problem is NP-complete on cycles and trees.

Summing up the above results, the only cases that remain open for paths and cycles are the multiple capacity preemptive vehicle routing problems on paths and cycles. (Note that the multiple capacity nonpreemptive routing problem on trees is NP-complete, as an instance of a unit capacity nonpreemptive routing problem on trees can be easily reduced to a multiple capacity nonpreemptive routing problem on trees.)

The multiple capacity preemptive routing problem on paths was investigated by Karp [7] and Guan [2]. Karp constructed a polynomial time algorithm for this problem under two constraints: (1) the starting vertex is an endpoint of the path; and (2) the number of objects located at each vertex before and after the transportation should be the same. In [2], Guan analyzed the time complexity of Karp's algorithm and showed that the algorithm takes $O(kn \log\log n)$ time, where k is the number of objects and n is the number of vertices. Then Guan constructed an algorithm with time complexity $O(k+n)$ and which does not require that the number of objects located at each vertex before and after the transportation be the same. However, it is crucial for both Karp's and Guan's algorithms that the starting vertex is an end vertex of the path.

In this paper, we give a polynomial time algorithm for the multiple capacity preemptive routing problem on paths, without imposing any constraints. The time complexity of our algorithm is $O(kn+n^2)$. Then we go on to show that our algorithm can actually be applied to the multiple capacity preemptive routing problem on cycles, under the constraint that for each object the direction of transportation is given. (Note that to transport an object from vertex i to vertex j on a cycle, we can go in either the clockwise or counterclockwise direction.) We remark that in practice, an object is usually transported along the shorter arc connecting the two vertices. Thus the constraint is not very unnatural. However, there are cases in which we can reduce the cost by allowing some objects to go along the longer arc, connecting their initial vertices and destination vertices, even if the problem is a unit capacity version [1].

2. Preliminaries. An instance of a vehicle routing problem consists of an edge weighted graph G, a set O of objects together with their initial vertices and destination vertices, a designated vertex s of G which is the starting vertex as well as the ending vertex of the vehicle, and a constant c which is the capacity of the vehicle. The weight of an edge represents the distance or the cost of transporting the objects between the two vertices of the edge and which is assumed to be nonnegative. We shall assume throughout the paper that the number of vertices of the graph G is n and that the number of objects is k. We shall concentrate on the case where the graph G is a path whose vertices are labeled $1, 2, \ldots, n$, where $(i, i+1)$, $i = 1, 2, \ldots, n-1$ are the edges of G. The k objects will also be labeled by $1, 2, \ldots, k$. The initial vertex and destination vertex of object j are u_j and v_j, respectively. We represent an object j by a directed edge from u_j to v_j with label j.

Thus an instance of a vehicle routing problem can be represented by a mixed graph which has undirected as well as directed edges. Each undirected edge has a nonnegative weight, and each directed edge has a label from the integer set $\{1, 2, \ldots, k\}$.

A vertex is *useful* if it is the start vertex s, the initial vertex u_j, or the destination vertex v_j for some object j. It is clear that, if the end-point of the path is not a useful vertex, then it can be deleted from the path. An intermediate vertex of the path that is not a useful vertex can be eliminated from the path by replacing the vertex and its two adjacent edges by one edge with weight the sum of the weights of the two edges. Therefore, we assume that all vertices are useful vertices.

A *move* from a vertex x to a vertex y of the vehicle carrying a set of objects Z is designated by $(x, y; Z)$. Each move $(x, y; Z)$ with $Z \neq \emptyset$ is called a *carrying move*, otherwise, it is called a *noncarrying move*. Let c be the capacity of the vehicle. A carrying move with $|Z| = c$ is called a *full-carrying move*.

An object j is transported from x to y by a move $(x, y; Z)$ if j is at vertex x before the move, and $j \in Z$. After the move, the object j will be at vertex y.

A *transportation*, Q from v_0 to v_r, is a sequence of moves

$$Q = (v_0, v_1; Z_1), (v_1, v_2; Z_2), \ldots, (v_{r-1}, v_r; Z_r).$$

Let Q be a transportation. Let $Q(j)$ be a subsequence of moves obtained from Q by deleting those moves that do not involve the object j. An object j is transported from x to y by a transportation Q, if $Q(j)$ is a transportation from x to y. If objects cannot be dropped at the intermediate vertices, then $Q(j)$ must be a consecutive subsequence of moves of Q.

A transportation is *valid* if $v_0 = v_r = s$, and each object is transported from its initial vertex to its destination vertex by Q. Unless stated otherwise, we consider only valid transportations, and we should use transportation instead of valid transportation.

The cost of the transportation, $c(Q)$, is the distance the vehicle traversed. That is,

$$c(Q) = \sum_{i=1}^{r} d(v_{i-1}, v_i),$$

where $d(v_{i-1}, v_i)$ is the sum of the weights of the edges from v_{i-1} to v_i in the underlying graph.

A transportation Q is an *optimal transportation* if $c(Q)$ is minimum among all valid transportations. The vehicle routing problem is to find an optimal transportation.

Let L be the underlying graph, which is a path. Denote $L[l, r]$ the subgraph of L induced by the vertices $l, l+1, \ldots, r$. Let $L[u, v]$ and $L[x, y]$ be two disjoint subgraphs of L. Define $f([u, v], [x, y])$ to be the number of objects with initial vertices in $L[u, v]$ and destination vertices in $L[x, y]$.

Since the underlying graph is a path and the vehicle must return to s, each edge must be traversed by the vehicle at least twice, once in each direction. Furthermore, for each edge (u, v), the number of times the vehicle traverses the edge from u to v must be equal to the number of times the vehicle traverses the edge from v to u. We therefore define a balanced problem as follows. For each edge $e = (v, v + 1)$, let

$$\lambda_e = \max\{\lceil f([1, v], [v + 1, n])/c \rceil, \lceil f([v + 1, n], [1, v])/c \rceil, 1\}.$$

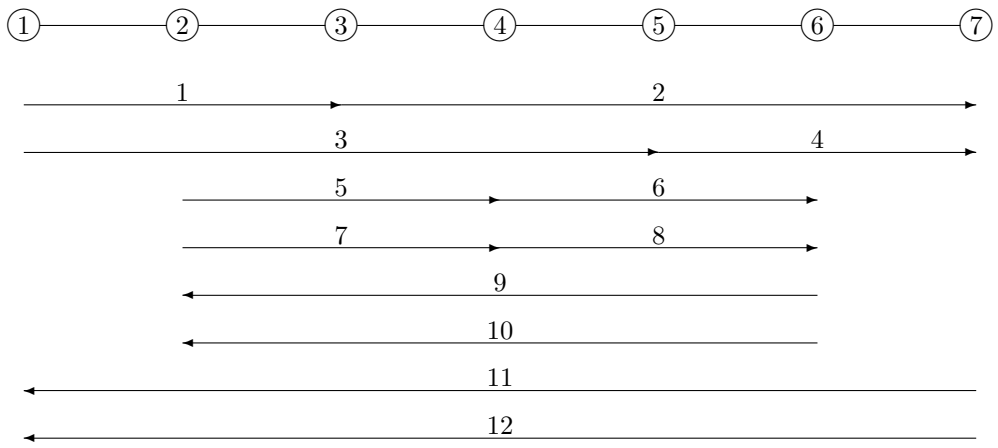


FIG. 2.1. *An example of the vehicle routing problem.*

That is, λ_e is the minimum number of times that the vehicle must traverse the edge e in either direction. A problem with $f([1,v],[v+1,n]) = f([v+1,n],[1,v]) = c\lambda_e$, for each edge $e = (v, v+1)$, $v = 1, 2, \ldots, n-1$, is called a *balanced problem*.

Given an arbitrary instance of the vehicle routing problem, one can add objects to make it into a balanced problem without increasing the cost of an optimal transportation. A trivial method is to add $c\lambda_e - f([1,v],[v+1,n])$ objects to be transported from v to $v+1$, and add $c\lambda_e - f([v+1,n],[1,v])$ objects to be transported from $v+1$ to v, for each edge $e = (v, v+1)$. It is clear that the total number of objects added to the problem is at most $((c-1)+(k+(c-1)))n_1 + 2cn_0$, where n_1 is the number of edges that are to be traversed by at least one object, and n_1 is the number of edges that are to be traversed by no objects. It can be done in $O(kn)$ time, assuming that c is a constant. For the remainder of this paper, we shall only consider balanced problems.

Figure 2.1 is an example of a vehicle routing problem, where the weight of each undirected edge is 1 and the capacity of the vehicle is 2. For clarity, directed edges are drawn with vertical offsets.

If the starting vertex is 1, then it is easy to verify that the following sequence of moves is a transportation of the given problem: $(1,2;\{1,3\})(2,4;\{5,7\})(4,6;\{6,8\})$ $(6,2;\{9,10\})(2,3;\{1,3\})(3,5;\{2,3\})(5,7;\{2,4\})(7,1;\{11,12\})$. Note that this transportation consists of only full-carrying moves, and each object is transported in such a way that it traverses the edges between its initial vertex and destination vertex exactly once, and traverses no other edges. Therefore, this is an optimal transportation. Indeed, it was proved in [2] that for any balanced multiple capacity preemptive vehicle routing problem on a path, there is a transportation that consists of only full-carrying moves. For the application in this paper, we shall briefly describe the algorithm given by Guan, which is divided into two steps [2]:

1. First we use a greedy algorithm, which starts at 1 and which arbitrarily picks c objects whose destinations are on the same side of the current vertex and moves the objects toward their destinations. The vehicle repeatedly moves objects in this direction. It changes the direction of movement only if, at

the current vertex, there are not enough objects to be moved in the current direction. It was proved in [2] that if there are not enough objects to be moved in either direction, then the current vertex must be the starting vertex. In other words, the route of the vehicle is a closed walk of the underlying graph.

Since the problem is balanced, when the above procedure stops, none of the remaining objects will be transported through the first edge. In other words, the first vertex is not a useful vertex to the remaining problem. Let x be the first vertex which is a useful vertex to the remaining problem. Then we start from vertex x, and apply the above greedy algorithm again, to obtain another sequence of moves, which form another closed walk of the underlying graph. Repeat this process to obtain a sequence of closed walks that together move every object to its destination.

2. As the second step, we "cut" and "paste" these closed walks to form a single closed walk. This is very similar to the algorithm that finds an Euler tour of an Eulerian graph. This single closed walk corresponds to a transportation which consists of only full-carrying moves.

We remark that the algorithm above does not work if the starting vertex is not an end vertex of the path. Indeed, in example 1, if the starting vertex $s = 4$, then there is no transportation in which all moves are full-carrying moves. We shall show in the following section how to compute an optimal transportation for the case where the starting vertex is not at the end-point of the path.

3. Preemptive routing in paths. In this section, we present an $O(kn + n^2)$ time algorithm to solve the multiple capacity preemptive vehicle routing problem on paths. We shall first establish a necessary and sufficient condition for a balanced problem to have a transportation that consists of only full-carrying moves.

THEOREM 3.1. *Let P be a balanced instance of a multiple capacity vehicle routing problem on a path, in which the starting vertex of the vehicle is s. There is a transportation that consists of only full-carrying moves if and only if there exists a sequence of full-carrying moves from s to one of the end-points of the path.*

Proof. If P has a transportation Q that consists of only full-carrying moves and that starts from s, then of course there exists a sequence of full carrying moves from s to one of the end-points of the path.

Assume that there exists a sequence of full carrying moves, say Q_1, from s to one of the end vertices, say 1. Let P' be the problem resulting from P by performing the sequences of moves in Q_1, and let P'' be the problem obtained from P' by adding c objects with initial vertex s and destination vertex 1. We denote by $\bar{X}$ the set of added objects.

The problem P'' is "almost" balanced, except that there might be somes edges that will not be traversed by any object. For simplicity, we may assume that the problem P'' is indeed balanced. In case there are such edges that will not be traversed by any object, we only need to consider each subproblem separately, and at the final stage, to combine the subtransportation together. This is quite a routine process, and the same technique is also used in Guan's algorithm described in the previous section.

Now let $\bar{P}$ be the problem obtained from P'' by switching the initial vertex and destination vertex of each object. Thus the objects of $\bar{X}$ now have initial vertex 1 and destination vertex s. Apply the algorithm described in the previous section to $\bar{P}$ in such a way that the very first step moves $\bar{X}$ from 1 to s. Also, when applying the second step of the algorithm to combine subtransportations into a single transportation, we shall avoid splitting the move $(1, s; \bar{X})$. This is possible because objects in

$\bar{X}$ are added to make the problem balanced, and hence the vertices between s and 1 must be traversed by the vehicle transporting objects not in $\bar{X}$.

Let the optimal transportation obtained by the algorithm be $\bar{Q}'$. Let $\bar{Q}$ be the inverse of $\bar{Q}'$, which is defined as the inverse of the sequence of the moves in Q and each move $(x, y; Z)$ is transformed into $(y, x; Z)$. It is clear that $\bar{Q}$ is a valid transportation of P'' that consists of only full-carrying moves.

An optimal transportation for the original problem, starting and ending at vertex s, can be obtained by concatenating Q_1 and $\bar{Q}$, and then deleting the last move, which is the move for the objects in $\bar{X}$. □

Now we present an algorithm that determines whether or not there exists a sequence of full-carrying moves from s to 1 or n.

We say a vertex x of the path is *reachable* (from s) if there is a sequence of full-carrying moves from s to x.

Suppose x is a reachable vertex. We define the *predecessor* $p(x)$ of x as follows: If $x \geq s$, then $p(x)$ is the smallest vertex such that $p(x) \leq s$ and any sequence of full-carrying moves from s to x passes through $p(x)$. If $x < s$, then we define $p(x)$ to be the largest vertex such that $p(x) \geq s$ and any sequence of full-carrying moves from s to x passes through $p(x)$.

Intuitively, if there is a sequence of full-carrying moves from s directly to x, then $p(x) = s$. Otherwise, to have a sequence of full-carrying moves from s to x, the vehicle may need to visit the vertices beyond s in order to pick up the necessary objects. The predecessor $p(x)$ is the vertex closest to s such that there exists a sequence of full-carrying moves from s to x in which the vehicle makes its last turn before reaching x.

Note that the predecessor function $p(x)$ is monotone, in the sense that if $x \leq y \leq s$, then $p(x) \geq p(y)$, and if $x \geq y \geq s$, then $p(x) \leq p(y)$. This monotone property reduces the computational complexity in finding an optimal transportation.

Suppose x is a reachable vertex. We define a function $c_x : E \to Z^+$, which is the smallest number of times the vehicle traverses the edge e, in the direction away from s, before it can reach the vertex x. The number of times the vehicle traverses the edge e in the direction toward s will not be counted in $c_x(e)$.

The values of $c_x(e)$ are computed as follows. Initially, let $c_s(e) = 0$ for every edge $e \in E$. Suppose $p(x)$ is known and that $c_{p(x)}(e)$ is determined for each edge e. The value of $c_x(e)$ can then be computed as follows. For $x \neq s$, $c_x(e) = c_{p(x)}(e) + 1$ if the edge e is between the two vertices s and x; otherwise, $c_x(e) = c_{p(x)}(e)$.

It can be proved by induction on $\sum_{e \in E} c_x(e)$ that for each reachable vertex x, there is a sequence of full-carrying moves from s to x that traverses the edge e in the direction away from s exactly $c_x(e)$ times, and any other full-carrying moves from s to x traverse the edge e in the direction away from s at least $c_x(e)$ times. Therefore, the vehicle must traverse each edge e $c_x(e)$ times, in the direction away from s, before reaching x.

The algorithm that determines the reachable vertex, the predecessor $p(x)$, and the value of $c_x(e)$ for each reachable vertex x is described as follows.

The algorithm keeps track of a set R, which is the set of currently known reachable vertices. By the definition of a reachable vertex, the induced subgraph $P[R]$ must be a connected subpath of P, which includes the start vertex s. We shall use the notation $R = [l, r]$ to denote the set of reachable vertices $l, l + 1, \ldots, r$.

Initially, $R = [s, s]$, $p(s) = s$, and $c_s(e) = 0$ for every $e \in E$. Suppose that, at the ith iteration, $R = [l, r]$. Then in the $(i + 1)$th iteration, the algorithm checks whether or not the vertices $l - 1$ and $r + 1$ are reachable. The testing of whether or not the

vertex $l-1$ is reachable is described as follows. For each vertex $x = p(l), p(l)+1, \ldots, r$, the algorithm decides whether or not there is a sequence of full-carrying moves from s to x and then directly from x to $l-1$ (i.e., without making another turn). We should prove that there is such a sequence of full carrying moves if and only if the following conditions are satisfied.

1. For each edge $e = (a, b) \in E$ between s and $l-1$, we have
$$c \cdot (c_x(e) + 1) \leq f([b, x], [1, a]).$$
2. For each edge $e = (a, b) \in E$ between s and x, we have
$$c \cdot c_x(e) \leq f([b, x], [1, a]).$$

If the inequalities above hold for all the edges e between x and $l-1$, then the algorithm adds $l-1$ to R, i.e., change R from $[l, r]$ to $[l-1, r]$, set $p(l-1) = x$, set $c_{l-1}(e) = c_x(e) + 1$ for those edges e between s and $l-1$, and set $c_{l-1} = c_x(e)$ for other edges e.

If the inequalities above do not hold for some edge e between x and $l-1$, then the vertex $l-1$ cannot be reached before the vehicle going further to the right of x. Thus the algorithm increases the value of x, provided that $x < r$, and then repeats the procedure above. If $x = r$, the vertex $l-1$ cannot be reached at the moment, and the algorithm starts checking whether or not $r+1$ is a reachable vertex. The method for checking whether or not $r+1$ is a reachable vertex is similar to the case for the vertex $l-1$.

The algorithm repeatedly checks whether or not the end-points of R can be extended, and the algorithm stops when neither $l-1$ nor $r+1$ is reachable.

We shall show that the algorithm above indeed finds all reachable vertices.

THEOREM 3.2. *Let P be a balanced problem. At the end of the algorithm, the interval R contains all reachable vertices. In other words, for each $x \in R$, there is a sequence of full-carrying moves from s to x, and for any $x \notin R$, there does not exist such a sequence of full-carrying moves.*

Proof. For simplicity, we may assume that $x \leq s$. We shall prove by induction the following stronger statement:

> For each $x \in R, x \leq s$, there is a sequence of full-carrying moves from s to x, in which the vehicle traverses each edge e exactly $c_x(e)$ times along the direction away from s, and in which $p(x)$ is the largest vertex that the vehicle passes through before reaching x. Moreover, for any other sequence of full-carrying moves from s to x, the vehicle traverses each edge e at least $c_x(e)$ times along the direction away from s, and the vehicle must pass through $p(x)$ before reaching x.

For $x > s$, a corresponding result holds. Although in our argument we only consider vertices $x \leq s$, our induction hypothesis is for all vertices added to R before the vertex x is added.

At the initial step, $R = [s, s]$, and the statement above is obviously true. Suppose x is added to R at step i. Then $p(x) \geq s$ is added to R at some previous step, and hence the statement above is true for any vertex y between s and $p(x)$ by the induction hypothesis. Thus there is a sequence of full-carrying moves from s to $p(x)$ that traverses each edge e exactly $c_{p(x)}(e)$ times along the direction away from s. Now we shall extend such a sequence by adding some full-carrying moves from $p(x)$ to x (without making turns). Of course, the only concern is whether or not there are

enough objects to be picked up and moved along the way. There are such objects for the following two reasons:

1. The previous sequence of moves traverses each edge $e = (a, b) \in E$ between s and $p(x)$ exactly $c_{p(x)}(e) - 1$ times in the direction from $p(x)$ to x (note that for these edges, this is the direction toward s, and that the vehicle starts at s) and traverses each edge $e = (a, b) \in E$ between s and x exactly $c_{p(x)}(e)$ times in direction from $p(x)$ to x.
2. For each edge $e = (a, b) \in E$ between s and $p(x)$, we have that

$$c \cdot c_{p(x)}(e) \leq f([b, p(x)], [1, a])$$

holds for each edge $e = (a, b) \in E$ between s and $p(x)$, and for each edge $e = (a, b) \in E$ between s and x, we have

$$c \cdot (c_{p(x)}(e) + 1) \leq f([b, p(x)], [1, a]).$$

Next we show that each sequence of full-carrying moves from s to x must pass through $p(x)$. Assume to the contrary that there is a sequence of full-carrying moves to x and that the maximum vertex that the vehicle passes through is $y < p(x)$. Then an initial segment of this sequence forms a subsequence of full-carrying moves from s to y. By the induction hypothesis, for each edge $e = (a, b) \in E$ between s and x, this subsequence of moves traverses e at least $c_y(e)$ times along the off s direction, and for each edge $e = (a, b) \in E$ between y and s, this subsequence of moves traverses e at least $c_y(e) - 1$ times along the toward s direction. On the other hand, it follows from the algorithm that, since $p(x) > y$, there is either an edge $e = (a, b) \in E$ between s and x such that

$$c \cdot (c_y(e) + 1) > f([b, y], [1, a]),$$

or there exists an edge $e = (a, b) \in E$ between s and y such that

$$c \cdot c_y(e) > f([b, y], [1, a]).$$

In either case, there are not enough objects to be picked up on the way for the vehicle to pass through the edge e. Therefore, every sequence of full-carrying moves from s to x pass through $p(x)$.

The argument above can be easily modified to show that every sequence of full-carrying moves from s to x traverses each edge e at least $c_x(e)$ times along the off s direction. We shall omit the details.

Also, the argument above can be modified to show that for any vertex $x \notin R$, there exists no sequence of full-carrying moves from s to x, and we shall omit the details. □

As shown by example 1, for some balanced problems, it is possible that there does not exist a transportation which consists of only full-carrying moves. By Theorem 3.1, for such a problem, there does not exist a sequence of full-carrying moves from s to an end-point of the path. In other words, the vertices 1 and n are not reachable.

For such problems, the optimal transportation must contain moves $(x, y; Z)$ with $|Z| < c$. It is straightforward to verify that, for balanced problems, we may restrict ourselves to full-carrying moves and noncarrying moves. To determine a minimum transportation cost for a problem, it suffices to determine the minimum distance traveled by the vehicle with noncarrying moves. As each edge must be traversed by

the vehicle the same number of times in both directions and, as the original problem is balanced, we conclude that for each edge e, the number of times the vehicle traverses e in the two directions with a noncarrying move is equal. Thus we may just count the number of times the vehicle traverses the edge e in the off s direction with a noncarrying move.

We define the cost of a noncarrying move to be the distance of that move. Suppose there is a vertex x of the path which is not reachable. We define $c(x)$ to be the minimum cost reaching x, i.e., the minimum total distance that the vehicle needs to travel with noncarrying moves before reaching x. We shall present an algorithm that determines recursively the values of all $c(x)$.

Before we present the algorithm, we consider again the example shown in Figure 2.1. Assume that the capacity of the vehicle $c = 2$. If the start vertex $s = 6$, then all of the vertices are reachable vertices. However, if the start vertex $s = 4$, then the reachable set $R = [2, 6]$. Readers are advised to try to find the minimum cost of adding objects to make each of the other vertices reachable.

In the algorithm, we keep track of the following parameters:

S: the set of vertices whose optimal cost has already been computed.

v: the last vertex added to S.

$c(x)$: the current known minimum cost of reaching x.

$c_x(e)$: the current known minimum number of times the vehicle traverses the edge e, in the direction away from s, in order to reach the vertex x with the cost $c(x)$.

$\epsilon_x(e)$: the currently known minimum number of times the vehicle crosses the edge e in the off s direction, with a noncarrying move, in order to reach x.

Our algorithm is similar to Dijkstra's algorithm for finding the shortest path between two vertices in a weighted graph. The above parameters will be updated at each iteration. Note that S must be a connected subgraph of the path; hence $S = [l, r]$ is always an interval.

Initially, we set $S = [s, s]$, $v = s$, $c(s) = 0$, $c_s(e) = 0$, and $\epsilon_s(e) = 0$ for every $e \in E$. For each $x \neq s$, we let $c(x) = \infty$, $c_x(e) = \infty$, and $\epsilon_x(e) = \infty$ for every edge e.

At each iteration, the parameters are updated by the following rules:

1. If $v \leq s$, then for each vertex $x \notin S$ and $x > s$, we calculate the cost $c'(x)$ of moving the vehicle from s to x in such a way that the vehicle reaches v first (with cost $c(v)$) and then goes directly from v to x (with some additional cost that can be calculated easily). Then we compare the new cost $c'(x)$ with the current value $c(x)$ and replace it with $c'(x)$ if $c'(x) < c(x)$.
2. If $v \geq s$, then we use the value of $c(v)$ to update the cost of $c(x)$ for all $x < s$, in the same way.

After we finish updating the cost function, we then add a vertex x with minimum cost $c(x)$ among the vertices $V - S$ into S, and then repeat the above process again.

This is exactly what was done in Dijkstra's algorithm for finding the shortest path between two vertices in a weighted graph. However, we need some tools to calculate the cost $c'(x)$.

In order to calculate the additional cost of moving the vehicle from v to x, we need to know how many times the vehicle traverses each edge e along the direction from v to x. Then we determine if there are enough objects to be transported by the vehicle to cross the edge. In case there are not enough objects to be transported by the vehicle to cross the edge $e = (a, b)$, then the vehicle needs to cross the edge e with a noncarrying move, and that will contribute to the cost of reaching x.

As noted before, we may just count the number of times the vehicle traverses the

edge e in the off s direction with a noncarrying move. We shall not consider the cost of moving the vehicle toward s.

Let $g_{x,v}(e)$ be the minimum number of times the vehicles traverses the edge e in the direction from v to x under the condition that the vehicle first reaches v and then directly goes to x. Then since $c_v(e)$ is the minimum number of times the vehicle traverses the edge e in the off s direction in the process of going from s to v, we conclude that:

1. for edges e between v and s, $g_{v,x}(e) = c_v(e)$; and
2. for edges e between s and x, $g_{v,x}(e) = c_v(e) + 1$.

Assume that $v \le s < x$. Let $W(v,x)$ be the set of edges $e = (a,b)$ between s and x such that $f([v,a],[b,n]) < c \cdot (g_{v,x}(e) - \epsilon_v(e))$. Then it is clear that $W(v,x)$ is the set of edges that the vehicle needs to traverse with a noncarrying move, along the way from v to x. (Note that $f([v,a],[b,n])$ is the total number of objects that can be transported from a to b at the present time; $g_{v,x}(e)$ is the total number of times the vehicle traversed from a to b up to now, including the trip from v to x; and $\epsilon_v(e)$ is the total number of times the vehicle has already traversed from a to b with noncarrying moves.)

Thus we set $c'(x) = c(v) + \sum_{e \in W(v,x)} w(e)$. Then we compare $c'(x)$ with $c(x)$. If $c'(x) < c(x)$, we do the following: set $c(x) = c'(x)$, set $\epsilon_x(e) = \epsilon_v(e)$ for $e \notin W(v,x)$ and $\epsilon_x(e) = \epsilon_v(e) + 1$ for $e \in W(v,x)$, set $c_x(e) = c_v(e)$ for each e between v and s and $c_x(e) = c_v(e) + 1$ for each e between x and s.

Find a vertex x in $V - S$ such that $c(x) = \min\{c(y) : y \in V - S\}$, and then let $S = S \cup \{x\}$. Then we set $v = x$ and repeat the procedure above.

The algorithm terminates when one of the end vertices of the path is added to S. Without loss of generality, we assume that 1 is added to S. Then $2c(1)$ is the minimum distance that must be traveled by the vehicle with noncarrying moves in any transportation for the problem.

THEOREM 3.3. *The algorithm described above indeed determines the minimum distance that must be traveled by the vehicle with noncarrying moves in any transportation for the problem.*

Proof. First we note that there is a transportation for the problem in which the total distance traveled by the vehicle with noncarrying moves is $2c(1)$. Indeed, in the process of reaching 1 by taking some noncarrying moves, we may replace each noncarrying move that crosses an edge $e = (a,b)$ in the off s direction by the addition of c objects from a to b, and c objects from b to a. Then after these objects are added, the resulting problem is still balanced and in this problem the vertex 1 becomes a reachable vertex. Then by Theorem 3.1, there is a transportation for this new problem that consists of only full-carrying moves. By omitting those added objects, we obtain a transportation of the original problem in which the total distance traveled by the vehicle with noncarrying moves is $2c(1)$.

Next we shall show that in any other transportation, the total distance traveled by the vehicle with noncarrying moves is at least $2c(1)$. Since in any transportation, the vehicle must reach vertex 1, it suffices to show for the vehicle to reach the vertex 1, it has to travel a distance of at least $c(1)$ in the off s direction, with noncarrying moves. We shall show by induction that for each vertex x of S, the vehicle needs to travel a distance of at least $c(x)$ in the off s direction, before reaching x.

Suppose S^* is the set of vertices of S which are added to S before x, and suppose to the contrary of the theorem that there is a sequence of moves, say Q_1, for the vehicle to reach x with less noncarrying move distance. By the induction hypothesis,

and by the procedure of updating the value of $c(x)$, it is straightforward to see that in the sequence of moves Q_1, the vehicle must reach some vertex $y \notin S^*$ before reaching x. Let y_0 be the first vertex not in S^* which is reached by the vehicle in this sequence of moves. By induction hypothesis and the procedure for defining $c(y)$ at that step, we conclude that the total distance traveled by the vehicle with noncarrying moves along the off s direction is at least $c(y) \geq c(x)$, which is a contradiction. Here $c(y)$ is the value at the step in which we add x into S. (Recall that in the algorithm the value $c(y)$ is updated at each step.) □

THEOREM 3.4. *The algorithm described above terminates in $O(kn + n^2)$ time, where k is the number of objects to be transported and n is the number of vertices in the path.*

Proof. After each iteration, one vertex is added to S. Hence there are at most n iterations. The crucial step to make the algorithm efficient is to compute the values of $f([v, x], [x+1, n])$ and $f([v, x+1], [1, x])$ efficiently. We shall show how to compute all of the values of $f([v, x], [x + 1, n])$, for $v = s, s + 1, \ldots, n$, $x = v, v + 1, \ldots, n - 1$, in $O(kn + n^2)$ time. The computation for $f([v, x + 1], [1, x])$ for $v = s, s - 1, \ldots, 1$, $x = v - 1, v - 2, \ldots, 1$ is done in a similar way.

For each vertex v, let $\alpha(v)$ be the set of objects with initial vertex v and with destination vertices to the right of v. For $v, x \in V$, $v < x$, let $\beta_v(x)$ be the set of objects with initial vertices between v and $x - 1$ and destination vertex x, and let $\delta(v, x)$ be the set of objects with initial vertex v and destination vertex x. By definition,

$$\beta_v(x) = \beta_{v+1}(x) \cup \delta(v, x).$$

For any given v, $v \leq s$, the value of $f([v, v], [v + 1, n]) = |\alpha(v)|$, and for $x = v + 1, v + 2, \ldots, n - 1$,

$$f([v, x], [x + 1, n]) = f([v, x - 1], [x, n]) + |\alpha(x)| - |\beta_v(x)|.$$

After balancing objects are added, there are at most $O(kn)$ objects. It is straightforward to compute all the values of $\delta(v, x)$, for all $v, x \in V$ in $O(kn + n^2)$ time. After these values are computed, for any given v, the values of $f([v, x], [x + 1, n])$, $x = v, v + 1, \ldots, n - 1$ can each be computed in constant time. Therefore, our algorithm can be implemented in $O(kn + n^2)$ time. □

An analysis of the algorithm shows that we have the term kn in the time complexity, simply for the reason that we may add $O(kn)$ objects to make it a balanced problem. We remark that with a complex algorithm, we can produce a balanced problem in which the number of added objects is at most $O(k + n)$. If we use that algorithm to produce the balanced version (which we did not use for the reason of simplicity), then the time complexity of our algorithm can be reduced to $O(k + n^2)$.

4. Preemptive routing on cycles. We shall show in this section that the algorithm in the previous section can be applied to the multiple capacity preemptive vehicle routing problems on cycles, provided that the arc along which each object is to be transported is determined. In practice, we usually transport an object along the shorter of the two arcs connecting its initial vertex and destination vertex.

We assume that the cycle has vertices $1, 2, \ldots, n$ embedded in the plane in this order along the clockwise direction. Furthermore, we assume that every vertex is a useful vertex, that is, it is either the start vertex s, or the initial vertex u_j, or the destination vertex v_j for some object j. A vertex that is not a useful vertex can be

eliminated from the cycle by replacing the vertex and its two adjacent edges by one edge with the weight the sum of the weights of the two edges.

Unlike the path, for a vehicle to start and end at a vertex s of a cycle, it is not necessary that the vehicle traverses each edge the same number of times in both directions. This is because the vehicle can go around the cycle in one direction many times and return to the staring vertex.

We shall employ techniques of [1] for overcoming this difficulty. First we modify the definition of a balanced problem. Let e be an edge of the cycle C, $\lambda(e)$ be the number of objects that must traverse the edge e in a clockwise direction, and $\lambda'(e)$ the number of objects that must traverse the edge e in a counterclockwise direction. A problem is balanced if and only if, for each edge e of C,

$$\lambda(e) \equiv \lambda'(e) \equiv 0 \bmod c$$

and

$$\phi(e) = \lambda(e) - \lambda'(e) = c\psi,$$

where c is the capacity of the vehicle and ψ is a constant.

The algorithm for finding an optimal tour is summarized as follows. First, find the balanced version of the problem for some fixed value of ψ. Using the algorithm for the paths, find a sequence of moves with minimum total distance of noncarrying moves such that the vehicle visit all the vertices of the cycle. Similarly we define $c(x)$ to be the minimum total distance of noncarrying moves to reach x. Let v be the vertex with largest cost $c(v)$ among all vertices in C. Then similar to Theorem 3.3, it can be proved that $2c(v)$ is the minimum total distance traveled by a vehicle with noncarrying moves in a transportation for the problem, under the condition that the vehicle goes around the cycle clockwise ψ times.

For each number ψ between $-\max_{e\in E}\{\frac{\lambda'(e)}{c} + 1\}$ and $\max_{e\in E}\{\frac{\lambda(e)}{c} + 1\}$, we apply the above algorithm. At the very end we choose the value of ψ for which the corresponding cost is minimum. That will be the minimum cost of a transportation for the problem.

We remark that this algorithm works only if each object is moved in the predetermined direction. The general problem remains open. In addition to this problem, we list two more open problems:

1. Suppose the underlying graph G is obtained from a fixed number, say k, of paths by joining one of their end points together. In other words, G is like a star, with a constant number of branches. What is the complexity of multiple capacity preemptive vehicle routing problems on G?
   We note that if the number of branches is not a constant, then the problem was shown to be NP-complete [2].
2. Suppose the capacity of the vehicle is unlimited; then what is the complexity of the vehicle routing problem on some special graphs such as trees or graphs of bounded tree-width?
   We note that this problem is easily seen to be polynomial on paths and cycles and NP-complete for general graphs.

REFERENCES

[1] M. J. Atallah and S. R. Kosaraju, *Efficient solutions to some transportation problems with application to minimizing robot arm travel*, SIAM J. Comput., 17 (1988), pp. 849–869.

[2] D. J. Guan, *Routing a vehicle of capacity greater than one*, Discrete Appl. Math., 81 (1998), pp. 41–57.
[3] G. N. Frederickson, *A note on the complexity of a simple transportation problem*, SIAM J. Comput., 22 (1993), pp. 57–61.
[4] G. N. Frederickson and D. J. Guan, *Preemptive ensemble motion planning on a tree*, SIAM J. Comput., 21 (1992), pp. 1130–1152.
[5] G. N. Frederickson and D. J. Guan, *Nonpreemptive ensemble motion planning on a tree*, J. Algorithms, 15 (1993), pp. 29–60.
[6] G. N. Frederickson, M. S. Hecht, and C. E. Kim, *Approximation algorithms for some routing problems*, SIAM J. Comput., 7 (1978), pp. 178–193.
[7] R. M. Karp, *Two combinatorial problems associated with external sorting*, in Combinatorial Algorithms, Courant Computer Science Symposium 9, R. Rustin, ed., Algorithmics Press, New York, 1972, pp. 17–29.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 603–623, November 1998

006

EDGE-CONNECTIVITY AUGMENTATION PRESERVING SIMPLICITY*

JØRGEN BANG-JENSEN† AND TIBOR JORDÁN†

Abstract. Given a simple graph $G = (V, E)$, our goal is to find a smallest set F of new edges such that $G = (V, E \cup F)$ is k-edge-connected and simple. Recently this problem was shown to be NP-complete. In this paper we prove that if OPT_P^k is high enough—depending on k only—then $OPT_S^k = OPT_P^k$ holds, where OPT_S^k (OPT_P^k) is the size of an optimal solution of the augmentation problem with (without) the simplicity-preserving requirement, respectively. Furthermore, $OPT_S^k - OPT_P^k \leq g(k)$ holds for a certain (quadratic) function of k. Based on these facts an algorithm is given which computes an optimal solution in time $O(n^4)$ for any fixed k. Some of these results are extended to the case of nonuniform demands as well.

Key words. edge-connectivity augmentation of graphs, network design, connectivity, combinatorial optimization

AMS subject classifications. 05C40, 05C85, 68R, 90B12

PII. S0895480197318878

1. Introduction.

In the last decade several graph augmentation problems have been investigated. The connectivity augmentation problems, especially, attracted considerable attention due to the various connections to the so-called network design problems which model the survivability problems of (telephone or computer, etc.) networks. In these problems a graph (or digraph) G and a target connectivity number k are given and the goal is to find a smallest set F of new edges which makes G k-edge-connected, that is, for which the augmented graph $G = (V, E \cup F)$ is k-edge-connected and $|F|$ is as small as possible. (Sometimes the goal is to increase the vertex-connectivity of G. In this paper we consider edge-connectivity problems only.) Note that the set F of new edges may contain parallel edges and edges which are parallel to edges of G as well.

The edge-connectivity augmentation problem and a number of its extensions can be solved efficiently. Since Watanabe and Nakamura's first polynomial time algorithm, several other efficient algorithms have been developed; see [4], [8], [22], [25] and also [3], [11], [21] for some important results. For a survey of this area see [9].

However, there are several versions of the connectivity augmentation problem which remain open. For example, in some cases the goal is to increase the connectivity by maintaing certain properties of the starting graph G. Depending on these properties, one obtains problems of a very different nature. Kant and Bodlaender [16] proved that the problem where the goal is to increase vertex-connectivity and the planarity of G is to be preserved is NP-complete. Nagamochi and Eades [19] solved some cases of the corresponding edge-connectivity problem in polynomial time. Hsu

*Received by the editors March 24, 1997; accepted for publication (in revised form) March 31, 1998; published electronically September 1, 1998. A preliminary version of this paper appeared in the *Proc. 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE Computer Society Press, Los Alamitos, CA, 1997, pp. 486–495.

http://www.siam.org/journals/sidma/11-4/31887.html

†Department of Mathematics and Computer Science, Odense University, DK-5230 Odense, Denmark (jbj@imada.ou.dk, tibor@imada.ou.dk). This research was supported by Danish Natural Science Research Council grant 28808 and in part by Hungarian National Foundation for Scientific Research grant OTKA T17580.

and Kao [13] showed how to increase a variant of vertex-connectivity while maintaining the bipartiteness of the graph in polynomial time. Recently, Bang-Jensen et al. [1] proved that edge-connectivity can be optimally increased in polynomial time preserving bipartiteness (or, in general, l-partiteness).

In this paper we deal with another property to be preserved: the simplicity of G. As it is stated in [9]: "It is an important open problem to find algorithms that do not add parallel edges." Partial results in this direction have been obtained by Frank and Chou [10], Naor, Gusfield, and Martel [22], Taoka, Takafuji, and Watanabe [23], and Watanabe and Yamakado [24] (the details are given below), but the complexity of the general problem was still open. Recently the second author proved that the simplicity-preserving k-edge-connectivity augmentation problem is NP-complete, even if the starting graph is already $(k-1)$-edge-connected; see [15] and [2]. On the other hand, as we shall prove, the problem becomes polynomially solvable if the target connectivity k is fixed. We give an $O(n^4)$ algorithm for this variant (where the running time depends exponentially on k).

Let $G = (V, E)$ be an l-edge-connected simple graph with $|V| \geq k + 1$. The *simplicity-preserving k-edge-connectivity augmentation problem* is to find a smallest set F of new edges which makes G k-edge-connected without creating parallel edges, that is, $G' = (V, E \cup F)$ must be a k-edge-connected simple graph and subject to this $|F|$ must be minimal. Such an F is called an optimal simple augmentation of G (with respect to k).

The very first paper that deals with a similar problem is from 1970 and is due to Frank and Chou [10]. They solve the simplicity-preserving edge-connectivity augmentation problem in the special case where the starting graph G has no edges. In this case they can handle nonuniform demands as well, where the edge-connectivity requirements may be different between different pairs of vertices. Besides the solution of this version — which is in fact a construction problem rather than an augmentation problem — there are some recent results which deal with small target connectivity values k or solve some very special case for general k.

Let us denote the size of a smallest k-edge-connected (k-edge-connected and simple) augmentation of a graph G by $OPT_P^k(G)$ (and $OPT_S^k(G)$, respectively). Obviously $OPT_P^k \leq OPT_S^k$ for any k and any graph G.

It can be checked easily that the linear-time 2-edge-connectivity augmentation algorithm of Eswaran and Tarjan [6] does not create parallel edges; thus it solves the simplicity-preserving version, too, for $k = 2$. It was proved in [24] that $OPT_S^k = OPT_P^k$ holds for $k = 3$ as well, by showing a polynomial algorithm which solves the 3-edge-connectivity augmentation problem optimally without creating parallel edges.

This is not the case in general. As it was observed already in [23], $OPT_S^k \geq OPT_P^k + 1$ may hold if $k \geq 4$; see Figure 1.1. On the other hand, it was also shown in [23] that $OPT_S^k \leq OPT_P^k + 1$ if $l + 1 = k$ with $k = 4$ or $k = 5$, and in these special cases we have $OPT_S^k = OPT_P^k$, provided that $OPT_P^k \geq 4$. In [22] it was observed that $OPT_S^k(G) = OPT_P^k(G)$ if $l + 1 = k$ and the minimum degree in G is at least k.

Besides the construction of a polynomial algorithm for any fixed k, our goal in this paper is to show that there exist polynomials f, g such that if $OPT_P^k(G) \geq f(k)$, then $OPT_S^k(G) = OPT_P^k(G)$, and $OPT_S^k(G) \leq OPT_P^k(G) + g(k)$ for any k and any graph G. These results are presented in section 3. Specializing our proofs to the case where $l + 1 = k = 4$, we give simpler proofs for (extensions of) some results of [23] in section 4. In section 5 we indicate how our main results can be extended to the case where local edge-connectivity requirements must be satisfied.

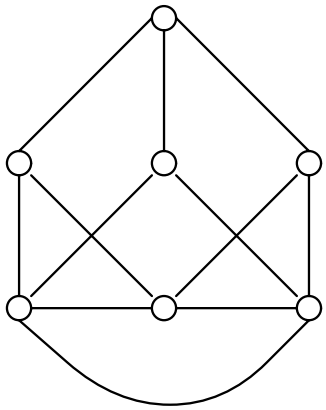


FIG. 1.1. $OPT^4_S(G) = 3$ *and* $OPT^4_P(G) = 2$.

2. Terminology and some basic results. In this section we first introduce the basic notation and definitions and then list those theorems and algorithms (mostly from Frank's paper [8]) which we shall use in our proofs. We assume that the reader is familiar with the results of [8].

Let $G = (V, E)$ be an undirected graph. For two disjoint subsets X and Y of V, the number $d(X, Y)$ denotes the number of edges between X and Y, and we define the degree of a subset X as $d(X) := d(X, V - X)$. A set consisting of a single vertex v is simply denoted by v. Thus $d(v)$ stands for the degree of v. The degree-function of a graph G' is denoted by d'. An edge connecting the vertices x and y is denoted by xy. Sometimes xy will refer to an arbitrary copy of the parallel edges between x and y, but this will not cause any confusion. Adding or deleting an edge e to/from a graph G is often denoted by $G + e$ or $G - e$, respectively. Adding or deleting a set Y of vertices to/from a set X of vertices is denoted by $X \cup Y$ or $X - Y$, respectively. For a set F of edges, $V(F)$ denotes the set of end-vertices of edges of F. The subgraph of G induced by a subset X of vertices is denoted by $G[X]$. The maximum degree of the graph G is $\Delta(G)$. For a vertex v we use $N(v)$ to denote the set of vertices adjacent to v. A *subpartition* of V is a collection of pairwise disjoint subsets of V.

The operation *splitting off* a pair vs, st of edges ($v \neq t$) from a vertex s means that we replace the edges vs, st by a new edge vt. A *complete splitting* from a vertex s (with even degree) is a sequence of $d(s)/2$ splittings of pairs of edges incident to s. A graph $G = (V, E)$ is *k-edge-connected* if

$$d(X) \geq k \quad \text{for all } \emptyset \neq X \subset V. \tag{1.1}$$

The *edge-connectivity* of G is the largest integer k for which G is k-edge-connected. The *local edge-connectivity* $\lambda(u, v)$ between two vertices u, v is the maximum number of pairwise edge-disjoint u-v paths.

The following equalities are well known.

PROPOSITION 2.1. *Let $G = (V, E)$ be a graph and $X, Y \subseteq V$. Then*

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X - Y, Y - X) \tag{1.2a}$$

$$d(X) + d(Y) = d(X - Y) + d(Y - X) + 2d(X \cap Y, V - (X \cup Y)). \tag{1.2b}$$

Let $G = (V + s, E)$ be a graph with a special vertex s such that (1.1) holds, that is, the edge-connectivity of G within V is at least k. We say that a pair of edges vs, st is an *admissible pair* if, after splitting off vs and st, condition (1.1) still holds. Otherwise vs, st form a *nonadmissible* pair. It is easy to see that vs and st are nonadmissible if and only if there exists a proper subset $X \subset V$ with $v, t \in X$ for which $d(X) \leq k + 1$. Such a set is called *dangerous*.

The following result of Lovász [17, Problem 6.53] — Theorem 2.2(a) below — is an important tool in [8]. Here we formulate a kind of extension, as well — part (b) of Theorem 2.2 — which will be used in some of our arguments. The proof follows easily from the proof of part (a) given in [8, pp. 35–36]. (Jackson [14] observed that a similar extension holds in the special case of Eulerian graphs.)

THEOREM 2.2. *Suppose that* (1.1) *holds in* $G = (V + s, E)$ *for some* $k \geq 2$ *and* $d(s) > 0$ *is even. Then*

(a) (see [17]) *for every edge* st *there exists an edge* su *such that the pair* st, su *is admissible.*

(b) *for every edge* st *the number of edges which are nonadmissible with* st *is at most* $k + 1$.

Proof. We prove part (b). Following Frank's proof of part (a) we observe that for every edge sv for which st and sv is a nonadmissible pair, the vertex v is either contained in a unique maximal dangerous set M containing t or contained in one of two maximal dangerous sets X, Y whose intersection contains t and for which $X \cup Y \neq V$, $d(s, X \cap Y) = 1$ hold. Since the edge sv contributes to the degree of M (or to the degree of X or Y), we obtain that in the former case there are at most k edges which are nonadmissible with respect to st, and in the latter case, using (1.2a) we get

$$2k + 2 \geq d(X) + d(Y) \geq d(X \cap Y) + d(X \cup Y) \geq k + d(X \cup Y),$$

which implies $d(X \cup Y) \leq k + 2$, from which $d(s, (X \cup Y) - t) \leq k + 1$ follows. □

Most of the results in this paper are based on Frank's algorithm [8] which solves the augmentation problem without the simplicity requirement and uses the splitting off operation as the main tool. His algorithm does not find all the intermediate optimal augmentations between $l + 1$ and k but only an optimal k-edge-connected augmentation. The other previously mentioned algorithms either use a one-by-one augmentation approach — like [3], [11], [22], [25] — or are based on splitting off [4], [21]. (We remark that Cai and Sun [4] gave the first algorithm for this problem which was based on the splitting off method.)

We say that the *successive augmentation property* holds for a certain augmentation problem if, for any increasing sequence $k_1 < \cdots < k_m$ of target-connectivities, there exists an increasing sequence $F_1 \subset \cdots \subset F_k$ of solutions such that F_i is optimal with respect to k_i. For example this property holds for the edge-connectivity augmentation problem of graphs and digraphs (with uniform demands); see [5], [22], [25]. Since the successive augmentation property does not hold for simple augmentations (see Figure 2.1), using Frank's algorithm seems to be promising for attacking the general case.

We now describe Frank's algorithm [8] which gives an optimal solution for any given (not necessarily simple) graph $G = (V, E)$ and target-connectivity k, provided that there is no simplicity-preserving requirement.

Frank's algorithm.

Phase 1. Add a new vertex s to V and a set F of new edges between s and some

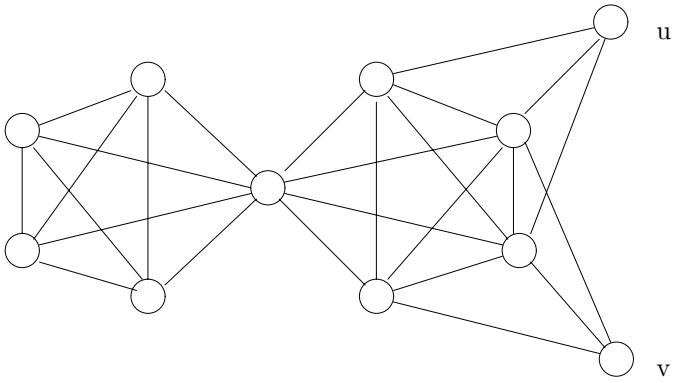


FIG. 2.1. *This graph G shows that the successive augmentation property does not hold. $OPT^4_S(G) = 1$ and $OPT^5_S(G) = 4$ for G. The edge $e = uv$ is the unique optimal augmentation with respect to $k = 4$, but $OPT^5_S(G + e) = 4$.*

vertices of V such that

$$\text{(1.1) holds in } G' = (V + s, E \cup F), \tag{2.1}$$

$$d'(s, v) \leq k \text{ for all } v \in V, \tag{2.2}$$

$$F \text{ is minimal (with respect to inclusion) subject to (2.1) and (2.2).} \tag{2.3}$$

Remark. It is easy to see that such an F exists. It was shown in [8] that $(*)$ there exists a subpartition $\mathcal{F} = \{X_1, ..., X_t\}$ of V for which $|F| = \sum_1^t (k - d(X_i))$ holds.

Phase 2. If $d'(s)$ is odd in G', add a new parallel edge between s and v for some $v \in V$ with $d'(s, v) \geq 1$.

Remark. In Frank's original algorithm, the extra edge which makes $d'(v)$ even can be added between s and any $v \in V$. However, this small modification in Phase 2 will be essential in our algorithm.

Phase 3. Split off admissible pairs of edges incident to s in arbitrary order, maintaining (1.1). When s becomes isolated, delete s.

Remark. In the third phase every edge can be split off by Theorem 2.2(a). The resulting graph is an optimal k-edge-connected augmentation of G since, after the first phase, $OPT^k_P(G) \geq |F|/2$ by $(*)$.)

3. Simplicity-preserving augmentation with uniform demands. Our idea is to modify the third phase of Frank's algorithm by introducing some additional rules which will determine the order of splittings. In certain cases not every admissible pair will be allowed to be chosen. As we shall see, this will make it possible to avoid the creation of parallel edges and hence to maintain simplicity provided that $d'(s)$ is high enough at the beginning of the third phase. Clearly, if we preserve simplicity during Phase 3, the resulting augmenting set will be optimal for the simplicity-preserving version, too.

If one wants to maintain simplicity while using Frank's algorithm, only those admissible pairs st, su may be split off for which t and u are not adjacent in the original graph and for which there is no new edge between t and u (that is, the pair st, su has not been chosen yet for another edge st and another edge su). We call such

an admissible pair *legal*. We say that a complete splitting from s is *feasible* if the resulting graph is simple and k-edge-connected.

Let $G = (V, E)$ be the starting graph and let $k \geq 2$ be the target-connectivity.

THEOREM 3.1. *Let $G' = (V + s, E \cup F)$ be the graph at the end of the second phase of Frank's algorithm and suppose that*

$$d'(s) \geq 3k^4. \tag{3.1}$$

Then there exists a feasible complete splitting from s.

Proof. Let S denote the set of neighbors of s. We claim that in the subgraph $H = G[S]$ induced by S each vertex has at most $k-1$ neighbors. To see this let us consider an edge $st \in F$. Since F is obtained from a minimal set $\bar{F}$ satisfying (2.1)–(2.3) by adding at most one edge e parallel to an edge of $\bar{F}$, there exists a set $X \subset V$ for which $t \in X$ and $d_{G'-e}(X) = k$. Each edge between t and some vertex in $S - X$ contributes to the degree of X, and each neighbor y of t in $S \cap X$ contributes to the degree of X by at least one by the existence of the edge sy. Thus

$$d_H(t) = d(t, S \cap (X - t)) + d(t, S - X) \leq d_{G'-e}(X) - 1 = k - 1. \tag{3.2}$$

Condition (2.2) implies that $|S| \geq 3k^3$. Since $\Delta(H) \leq k-1$, there exists (see the easy exercise [17, Problem 8.1]) a subset $T \subset S$ with $|T| = k^2 + 2k + 1 \leq 3k^3/k = 3k^2$ which is independent in H. Since such a T can be obtained by a greedy procedure, starting with an arbitrary vertex, we can assume that the end-vertex in V of the extra edge e added in the second phase of Frank's algorithm, if such an edge exists, belongs to T.

Our proof of the existence of a feasible complete splitting will follow from the analysis of the following algorithm which is a modified version of the third phase of Frank's algorithm. The goal is to split off only legal pairs until s becomes isolated.

After each splitting step we update S and H as follows. If we split off a pair st, su, then first the edge tu will be added to H. Then, if the edge st (or su) was the last copy of the edges between s and t (between s and u, respectively), then we delete t (u) from S and from H as well. Thus at every iteration a pair st, su of edges is legal if and only if t and u are not adjacent in the current H.

The algorithm has four parts, executed in the following order.

(1) Split off legal pairs of edges st, su of the form $t \in T$ and $u \in S - T$ until there are no parallel edges between s and any vertex of T (but keep one copy of each edge st, $t \in T$).

(2) Split off legal pairs of edges sv, su of the form $v, u \in S - T$ as long as possible.

(3) Split off all the edges su with $u \in S - T$ with an edge st, $t \in T$, for which st, su is a legal pair.

(4) Split off the remaining edges st, su (of the form $t, u \in T$).

Let us prove why this algorithm terminates with a complete feasible splitting. First observe that the proof of Theorem 2.2(b) combined with the proof of (3.2) implies the following statement.

PROPOSITION 3.2. *Let st be an edge, and let W be a subset of vertices of the current H not containing t. If $|W| \geq k + 2$, then st, su is a legal pair for some $u \in W$. Furthermore, if $|W| = k + 1$ and there is no legal pair st, su with $u \in W$, then $d'(s, t) = 1$.*

Proof. Let Z be a set of vertices of H for which $t \notin Z$ and there is no legal pair st, sz with $z \in Z$. Let $Z_1 \subseteq Z$ contain those vertices z' of Z for which the pair st, sz' is not admissible. Let $Z_2 = Z - Z_1$. By our assumption each vertex of Z_2 is adjacent to t

in H. If $Z_1 = \emptyset$, then by (3.2) we obtain $|Z| = |Z_2| \leq k-1$. Suppose $Z_1 \neq \emptyset$. It follows from the remarks we made in the proof of Theorem 2.2(b) that either there exists a unique maximal dangerous set M which includes $Z_1 \cup \{t\}$ or there exist two maximal dangerous sets X, Y with $t \in X \cap Y$ and $Z_1 \subset (X - Y) \cup (Y - X)$. Observe that each vertex $z \in Z_1$ ($z \in Z_2$) contributes to the degree of M or $X \cup Y$ by the existence of the edge sz (tz, respectively), and the edges from s to t have a positive contribution as well. Thus in the case of the unique M we get $|Z| = |Z_1| + |Z_2| \leq d'(M) - 1 \leq k$. In the case where we have two maximal dangerous sets X and Y we can count as in Theorem 2.2 and obtain $|Z| \leq d'(X \cup Y) - 1 \leq k + 1$. Thus $|Z| \leq k + 1$ follows and $|Z| = k + 1$ if only if $d(s, t) = 1$, as required. □

By (2.2) and the fact that Phase 2 adds at most one extra edge connecting s and T, we have to split off at most $(k-1)|T| + 1$ pairs in part (1). Proposition 3.2 and the inequality $|S - T| \geq 3k^3 - (k^2 + 2k + 1) \geq (k^2 + 2k + 1)(k - 1) + k + 2$ imply that this can be done, that is, we can always find legal pairs to split off following the rule of part (1). Therefore part (1) can be executed.

By Proposition 3.2 we obtain that at the end of part (2) the size of the current $S-T$ is at most $k+2$ and if $|S-T| = k+2$, then $d'(s, u) = 1$ for each $u \in S-T$. Thus using (2.2), it follows that in part (3) at most $\max\{k + 2, k(k + 1)\} = k(k + 1)$ edges must be split off, for which $k(k + 1) + k + 1$ vertices in T are sufficient to maintain feasibility by Proposition 3.2. Since $|T| = k^2 + 2k + 1 = k(k+1) + k + 1$, part (3) can also be executed.

At the beginning of part (4) the current H, induced by T, is an independent set with $d'(s, t) = 1$ for all $t \in T$. Hence H remains independent after an arbitrary sequence of splittings. Thus in part (4) we are allowed to split off admissible pairs in arbitrary order, which yields a (feasible) complete splitting by Theorem 2.2(a). □

We obtain the following corollary.

THEOREM 3.3. *If $OPT^k_P(G) \geq 3k^4/2$ for some graph G, then $OPT^k_S(G) = OPT^k_P(G)$ holds.*

If k is small, it is possible to sharpen the previous rough bound on $f(k) \leq 3k^4/2$. Using a more precise analysis we can even obtain the sharp value in the special case $k = 4$ (which was found already in [23], using a different approach). These details are given in section 4.

THEOREM 3.4. *$OPT^k_S(G) \leq OPT^k_P(G) + 2k^2 + 1$ for any starting graph $G = (V, E)$ and any target connectivity k.*

Proof. If $|V| \leq 4k - 4$, then $OPT^k_S \leq k(2k - 2)$ since for any $|V|$ there exist k-edge-connected simple graphs $H = (V, E(H))$ which are (almost) k-regular. Obviously $E(H)$ contains a set of edges which makes G k-edge-connected preserving simplicity. Thus we may assume that $|V(G)| \geq 4k - 3$.

The proof is based on a version of Frank's algorithm, where in the third phase certain edges will not be split off in pairs but will be replaced by one or two new edges using different operations. The first two phases are the same. At the beginning of Phase 3 we have a set F of new edges incident to s for which $OPT^k_P(G) = |F|/2$ holds. In Phase 3 first let us split off legal pairs as long as possible. By Proposition 3.2 the number of neighbors of s is at most $k + 2$ when no more legal pairs can be found and $d'(s) \leq k(k+1)$. In the rest of the procedure, instead of splitting off pairs of edges we take the remaining edges incident to s one by one and for such an sx we either delete it, without destroying (1.1), or, if it is not possible, we replace it by one or two edges on V, maintaining (1.1) and preserving simplicity. Let us focus on some edge $e = sx$ of the current G' in this last part. If e cannot be deleted without violating (1.1),

then there exist sets X', Y' with $x \in X' \subset V$, $d'(X') = k$ and $s \in Y' \subseteq V + s - x$, $Y' \cap V \neq \emptyset$, $d'(Y') = k$. Let us call such a set X' *x-tight* and such a set Y' *s-tight*.

PROPOSITION 3.5. *In G' there is a unique minimal x-tight set X and a unique minimal s-tight set Y, with respect to a fixed edge $e = sx$. Furthermore,* (1.1) *holds in $G' - e + e'$ for any edge $e' = x'y'$ with $x' \in X$ and $y' \in Y$.*

Proof. Let X be a minimal x-tight set and X' be an x-tight set which does not contain X. In this case, by (1.1) and (1.2b) we obtain

$$k+k = d'(X)+d'(X') = d'(X-X')+d'(X'-X)+2d'(X\cap X', V+s-(X\cup X')) \geq k+k+2,$$

a contradiction. This proves that X is unique. The uniqueness of Y can be proved similarly.

Let $x' \in X$ and $y' \in Y$ be two vertices and $e' = x'y'$ be an edge and suppose that (1.1) does not hold in $G'' = G' - e + e'$. Clearly, if a set $\emptyset \neq W \subset V$ violates (1.1), then W is x-tight in G'. Hence the uniqueness of X implies $x' \in W$ and then $y' \in W$ must also hold. From this it follows that $V + s - W$ is an s-tight set in G', which does not contain Y, a contradiction. This proves the proposition. □

Clearly, if X or $Y - s$ has size at least k, then there exist two vertices $x' \in X$ and $y' \in Y - s$ such that x' and y' are nonadjacent in G'. In this case, replacing sx by $e' = x'y'$ maintains (1.1) and preserves simplicity. Suppose that both X and $Y - s$ have size at most $k - 1$. There are at most $2k - 2$ vertices in $V - (X \cup Y)$ which are adjacent to x or to some vertex in $Y - s$, since each such vertex contributes to the degree of X or Y. We assumed $|V| \geq 4k - 3$; hence there exists a vertex $w \in V - (X \cup Y)$ which is neither adjacent to x nor to y for some vertex of $Y - s$. Then replacing e by xw and wy preserves simplicity and it is easy to see that it maintains (1.1) as well.

Thus substituting the remaining at most $k(k + 1)$ edges incident to s in G' by at most $2k(k + 1)$ edges, we obtain a solution with size at most $|F|/2 + 1.5k(k + 1) \leq OPT_P^k(G) + 1.5k(k + 1) \leq OPT_P^k(G) + 2k^2 + 1$. □

The graph K^*, defined as the disjoint union of two complete graphs K_{k+1} and $K_{k/2}$, connected by $k/2$ independent edges, shows that the biggest possible gap between OPT_S^k and OPT_P^k is indeed a quadratic function of k. (It is easy to check that $OPT_P^k(K^*) = (k/2+1)k/4$, but $OPT_S^k(K^*) = k^2/4$. Hence the difference in question is $k^2/8 - k/4$.)

Also note that the solution obtained by the algorithm of Theorem 3.4 has size at most $2|F|$, where F is the set of the new edges added by the first two phases. Since $|F|/2 \leq OPT_S^k(G)$, the previous method can also be considered as a 4-approximation algorithm (that is, an algorithm which gives a solution of size at most $4OPT_S^k(G)$) for the simplicity-preserving k-edge-connectivity augmentation problem, provided that $k < n/4$. Its running time is polynomial even if k is not fixed.

The basic idea of our algorithm for the simplicity-preserving k-edge-connectivity augmentation problem is the following. If OPT_P^k is large enough, we can simply follow the algorithmic proof of Theorem 3.1 which gives a solution of size $OPT_S^k = OPT_P^k$. If OPT_P^k is small, a trivial way of finding an optimal solution is to check every possible augmenting set with size less than $OPT_P^k + 2k^2 + 1$. By Theorem 3.4 we can find an optimal solution this way. However, although the number of such sets is a polynomial function of n for fixed k, the exponent still depends on k. To avoid this, we prove that when we check all the possible augmenting sets we may restrict the set T of possible end-vertices of the augmenting edges to a set of size $h(k)$ for an appropriate function h of k, and that such a T can be fixed in advance in constant time for any fixed k.

A set $X \subset V$ is *deficient* in $G = (V, E)$ (with respect to the target-connectivity k) if $d(X) < k$. A set $S \subseteq V$ is a *covering* of the deficient sets if $S \cap X \neq \emptyset$ for every deficient set X.

It follows from the correctness of Frank's algorithm — and also from Proposition 3.5 — that for every covering $S \subseteq V$ there exists an optimal solution F with $V(F) \subseteq S$ for the problem without simplicity requirement. This implies that if the optimum value is m, we can easily find a set S of vertices in G — the covering S, formed by the end-vertices of the new edges at the end of Phase 1 of Frank's algorithm will do — with size at most $2m$, such that there exists an optimal solution F with $V(F) \subseteq S$. Although in the simplicity-preserving version a covering S does not have this property in general (see the graph in Figure 1.1), we can find a relatively small subset T in this case, too, such that there exists an optimal solution F_S with $V(F_S) \subseteq T$.

To see this, let G be the starting graph and let S be a covering of the deficient sets of G. For each vertex y of G let us fix a set $L(y) \subseteq N(y)$ of vertices such that $|L(y)| = \min\{|N(y)|, 2k\}$. For each vertex $t \in S$ we define a subset $\bar{M}_t$ of vertices of V as follows. The vertex t itself belongs to $\bar{M}_t$ if and only if $d(t) \leq 2k-1$. A vertex $x \in V - t$ belongs to $\bar{M}_t$ if and only if there exists an xt-path P on at most k vertices for which $d(v) \leq 2k-1$ for every $v \in P - x$. Now let $M_t := \{t\} \cup L(t) \cup \bigcup_{v \in \bar{M}_t} (v \cup L(v))$. (It is not hard to see that $|M_t| \leq h(k) := 2k(2k-1)^k$ for any $t \in S$.) We claim that $T = \bigcup_{v \in S} M_v$ is a set with the required property. Note that $S \subseteq T$ holds and $|T| \leq |S|h(k)$. (The existence of such a set with size at most $|S|h'(k)$ for some function h' of k follows immediately from our previous results. However, to construct an efficient algorithm, we need to find such a set in advance without knowing an optimal solution.)

THEOREM 3.6. *There exists an optimal simplicity-preserving solution F with $V(F) \subseteq T$.*

Proof. For some edge $e = xy$ let $r(e) := |\{x, y\} \cap (V - T)|$. Let us choose an optimal simplicity-preserving solution F for which $r(F) := \sum_{e \in F} r(e)$ is as small as possible. If $r(F) = 0$, we are done. If $r(F) \geq 1$, there exists an edge $e = ab \in F$ for which at least one of its end-vertices is not in T. We may assume $b \notin T$. If we subdivide e by a new vertex s and then apply Proposition 3.5, we observe that in $G \cup F - e$ we have precisely two minimal deficient sets A, B, and for these sets we have $A \cap B = \emptyset$, $a \in A$, and $b \in B$. Furthermore, for any edge $e' = a'b'$ with $a' \in A$, $b' \in B$ the graph $G \cup F - e + e'$ is k-edge-connected. Thus it is enough to prove that there exists a vertex $t \in B \cap T$ which is not adjacent to a in $G \cup F$. Since $d_{G \cup F}(A, B - b) \leq d_{G \cup F}(A) - 1 \leq k - 1$, to prove that such a vertex t exists, it is sufficient to see that $|T \cap B| \geq k$. (Then replacing e by $e' = at$ would yield an optimal simple augmenting set F' with $r(F') < r(F)$, contradicting the choice of F.)

First let us prove that $S \cap C \neq \emptyset$ for every component C of the subgraph $G[B]$. Since B is deficient in $G \cup F - e$, it is deficient in G as well. Thus since S covers the deficient sets, $S \cap B \neq \emptyset$. If $S \cap C = \emptyset$ for some component C of $G[B]$, the set C is not deficient in G. Thus since each edge between C and $V - C$ contributes to $d(B)$, we obtain $d_{G \cup F - e}(B) \geq d_G(B) \geq d_G(C) \geq k$, a contradiction.

Now let $t \in S$ be a vertex in the component of $G[B]$ which contains b and let P be (the set of vertices of) a tb-path in $G[B]$. Let b' be the vertex of P closest to t on P which is not included in $\bar{M}_t$ and let P' denote the subpath of P with end-vertices t and b'. (Such a b' exists, since $b \notin \bar{M}_t$.) Then $P' - b' \subset M_t \subset T$. Thus if $|P' - b'| \geq k$, we are done, since $|T \cap B| \geq k$ follows. Assume now that $|P' - b'| \leq k - 1$. Then either $b' = t$ and hence $d(t) \geq 2k$, or, by the definition of $\bar{M}_t$ and b', there exists a

$z \in P' \cap \bar{M}_t$ with $d(z) \geq 2k$. In the former case we let $z = t$. Then we obtain that at least k vertices from $L(z)$ belong to B; otherwise $d_G(B) \geq d_G(z, V - B) \geq k$, a contradiction. Hence $|T \cap B| \geq |M_t \cap B| \geq |L(z) \cap B| \geq k$. □

The proofs of the previous three theorems lead to an algorithm whose running time is $O(n^4)$, provided that the target-connectivity k is fixed. In the rest of this section we sketch the algorithm and estimate the running time.

The input graph is $G = (V, E)$ with n vertices and e edges. First we add a new vertex s and construct a set F of new edges following Phases 1 and 2 of Frank's algorithm. Simultaneously we compute $OPT_P^k(G) = |F|/2$ and let $S = N(s)$ be our covering of the deficient sets. This can be done in time $O(e + kn^2)$ by applying algorithm "Augment" of Nagamochi and Ibaraki [21]. If $OPT_P^k(G) \geq 3k^4/2$, we proceed as described in Case II below. Otherwise we are in Case I, where $|S| \leq 3k^4$ and $OPT_S(G) \leq \bar{g}(k) := 3k^4/2 + 2k^2 + 1$ by Theorem 3.4. In this case we identify the set $T \subseteq V$ of vertices as defined before Theorem 3.6. From the algorithmic point of view, this can be done by computing a restricted BFS-tree from each vertex of S. Thus the number of steps we need to find T depends on k only. The last step of Case I is just a series of k-edge-connectivity tests. We check for each possible set of new edges F' of size at most $\bar{g}(k)$ (and with $V(F') \subseteq T$) whether $(V, E \cup F')$ is simple and k-edge-connected and choose the smallest good augmentation. Clearly, the number of possibilities is a function of k, and by Theorem 3.6 we find an optimal augmenting set. The number of steps in one of these tests is $O(e + kn^2)$ using the algorithm of [20].

Let us analyze Case II, where $OPT_S^k(G) = OPT_P^k(G)$ holds by Theorem 3.3. Following the proof of Theorem 3.1, first we identify a set T of vertices which is independent in $G[S]$ and has size $k^2 + 2k + 1$. This can be done by a greedy search in linear time. Then we follow the four parts of the modified third phase of Frank's algorithm: first we split off pairs of edges incident to s between T and $S - T$. For this we use the so-called s-based connectivity algorithm from [21] as a subroutine to test whether or not a pair is legal. One of these tests requires time $O(n(e + n))$ and the total number of tests in this first part depends on k only. In the second part we split off pairs whose end-vertices are in $S - T$ as long as possible. This requires at most $k(k + 2)n$ s-based connectivity tests. (There are at most k edges from each vertex $v \in S - T$ to s and by Proposition 3.2 after at most $k + 2$ tests we can find a legal pair including sv, if there is any.) The remaining two parts consist of some further s-based connectivity test: we split off all pairs between $S - T$ and T and then within T. These calculations imply the following theorem.

THEOREM 3.7. *The simplicity-preserving k-edge-connectivity augmentation problem can be solved in $O(n^4)$ time for any fixed k.*

The running time $O(n^4)$ hides a huge exponential function of k which makes the above described algorithm inefficient from any practical point of view. However, for those small values of k which may occur in applications, our results and methods can be substantially refined and the algorithm can be made efficient. This is the topic of the next section.

4. Augmenting from 3 to 4. In this section we give a full solution for the special case where G is 3-edge-connected and we want to make it 4-edge-connected. As we remarked, $k = 4$ is the smallest target value where OPT_P^k and OPT_S^k may be different. Our goal is to find the exact values of the functions f and g in this case. The proof implies an algorithm which does not use a series of 4-edge-connectivity tests as in Case I of the algorithm of the general case. The main result of this section,

Theorem 4.7, was already obtained by Taoka, Takafuji, and Watanabe in [23] — where most of the proofs are omitted — using a different approach which does not seem to work for arbitrary values of the target connectivity. We included this section to show how our method provides a fairly easy complete proof of this result.

A set X of vertices in a graph G with edge-connectivity l is called *critical* if $d(X) = l$ holds. The following easy lemma is left to the reader.

LEMMA 4.1. *Let G be simple and $(k-1)$-edge-connected but not k-edge-connected. Then every minimal critical set has size one or at least k.*

Lemma 4.1 shows that if X and Y are two disjoint critical sets, then there exist two vertices $x \in X$ and $y \in Y$ which are nonadjacent, unless both X and Y are singletons. This suggests that if the goal is to increase the connectivity by one, without creating new parallel edges, only those minimal critical sets which are singletons have an important role. The following lemma will make it possible to assume that every minimal critical set is a singleton.

LEMMA 4.2. *Let $G = (V, E)$ be a simple graph which is $(k-1)$-edge-connected but not k-edge-connected. Then there exists a $(k-1)$-edge-connected simple graph G^* for which*

(1) *every minimal critical set in G^* has size one,*
(2) $OPT_P^k(G^*) = OPT_P^k(G)$,
(3) $OPT_S^k(G^*) = OPT_S^k(G)$.

Proof. First note that for a minimal critical set X and a critical set Y either $X \subseteq Y$ or $X \cap Y = \emptyset$ holds by Proposition 2.1. This implies that the minimal critical sets of G are pairwise disjoint. Suppose that G contains some minimal critical sets $X_1, \ldots, X_r$ which are not singletons. By Lemma 4.1, each X_i has size at least k. Form a new graph $G^* = (V^*, E^*)$ from G by adding a new vertex x_i and joining it by $(k-1)$ edges to different vertices of X_i for each $i = 1, 2, \ldots, r$. We claim that G^*, which is easily seen to be $(k-1)$-edge-connected and simple, has properties (1)–(3) above. Clearly, the singleton critical sets in V and the new vertices $x_1, ..., x_r$ are minimal critical sets in G^*. Suppose that G^* has a minimal critical set X which is disjoint from all of these vertices. Such an X corresponds to a critical set in G and hence includes a minimal critical set X' of G. Now X' is not a singleton; hence by the construction of G^*, one of the new vertices is a neighbor of X. Thus X has degree more than $k-1$ in G^*, a contradiction. Hence it follows that the minimal critical sets in G^* are precisely the original singleton critical sets plus the vertices $x_1, \ldots, x_r$. This proves (1) and implies that during the first two phases of Frank's algorithm the same number of new edges are added to G and G^*, which gives (2). (Observe that at the end of the first phase of the algorithm, if applied to a $(k-1)$-edge-connected graph, there is precisely one new edge from the extra vertex s to each minimal critical set and there are no more new edges incident to s.) Now let us consider an optimal simple k-edge-connected augmentation F of G. Let us form a set F^* of new edges from F by replacing every edge $e \in F$ connecting two sets X_i and X_j by $x_i x_j$ and every edge $e' = xy \in F$ which enters some $x \in X_i$ (and $y \notin X_j$, $j = 1, ..., r$) by $x_i y$. It is easy to see that the graph $G^* + F^*$ is k-edge-connected and by the construction F^* contains no edges which are parallel to edges of G^*. Furthermore, since G^* is $(k-1)$-edge-connected, parallel copies of the edges of F^* can be deleted without destroying k-edge-connectivity. This gives $OPT_S^k(G^*) \leq OPT_S^k(G)$.

To prove the other inequality first observe that, since each X_i has size at least k, it follows that if $u \in V - X_i$ and ux_i is an edge in an optimal simple augmentation F^* of G^*, then there is at least one vertex in X_i which is not adjacent to u in G.

Similarly, if x_ix_j is an edge in F^*, then we can find vertices $u \in X_i$ and $v \in X_j$ so that uv is not an edge in G. Now define the following set of new edges F to be added to G: let F contain all those edges of F^* which connect vertices of V. Furthermore, for each edge of type $ux_i \in F^*$, where $u \in V - X_i$, let F contain an edge from u to a nonneighbor of u in X_i and, for each edge of type $x_ix_j \in F^*$, let F contain an edge uv such that $uv \notin E$ and $u \in X_i$, $v \in X_j$. We claim that $G'' = (V, E \cup F)$ is k-edge-connected which will imply that $OPT_S^k(G) \leq OPT_S^k(G^*)$. (By the construction, F does not contain edges parallel to edges of E. It is easy to see that F itself does not contain parallel edges, but to prove the inequality it is enough to observe that deleting one copy of two parallel edges in F does not destroy k-edge-connectivity.) Suppose that G'' is not k-edge-connected and let W be a set whose degree in G'' is $k-1$. Since G is $(k-1)$-edge-connected, W is critical in G as well and if $W \cap U \neq \emptyset$ for some minimal critical set U (in G), then $U \subseteq W$. Since none of the edges in F leave W we obtain that in G^* the set W^*, obtained by adding each of the vertices x_i for which $X_i \subseteq W$ to W, has degree $k-1$, a contradiction. This proves (3). □

In the next four lemmas we consider a graph $G' = (V + s, E \cup E' \cup F)$ obtained by applying the first and second phases and possibly some $p \geq 0$ iterations of the third phase of Frank's algorithm, starting from a 3-edge-connected (but not 4-edge-connected) simple graph $G = (V, E)$ with $|V| \geq 5$ and target $k = 4$. The edges E' are the edges obtained by the p splitting off operations. Our goal is to find $OPT_S^4(G)$ and an optimal augmenting set. By Lemma 4.2 we can assume that all neighbors of s in G' have degree 3 in G (i.e., they form singleton critical sets). Let $S = \{x \in V : sx \in F\}$ and let $H = G[S]$. We call a set $\emptyset \neq X \subset V$ *critical* in G' if $d'(X) = 4$ and *dangerous* if $d'(X) \leq 5$ holds. Since G is already 3-edge-connected, every dangerous set X has $d'(s, X) \leq 2$. Note that $d'(s, x) \leq 2$ for each $x \in V$ and $d'(s, x) = 2$ holds for at most one vertex of V. Therefore a vertex $x \in V$ with $d'(s, x) = 2$ exists if and only if $|S|$ is odd. We also obtain $E(H) \subseteq E(G)$. These facts are used several times in the following.

LEMMA 4.3. *The following holds for every $x \in S$.*

(1) *If $d_H(x) \geq 2$, then for every nonneighbor u of x in H, the pair sx, su is legal for splitting.*
(2) *There is at most one maximal dangerous set containing x and at least one other neighbor of s. In particular there is at most one nonneighbor y of x in H for which the pair sx, sy is not legal for splitting.*
(3) *Either $d_H(x) = 3$, or sx is in at most two pairs sx, su and sx, sv which are not legal for splitting.*

Proof. Suppose that $d_H(x) \geq 2$ and that X is a dangerous set containing x and some $u \in S$ for which $xu \notin E(H)$. Then the set $X - x$ has degree at most 2 in G, contradicting the fact that G is 3-edge-connected. This proves (1). To prove (2) suppose that x is contained in two maximal dangerous sets A and B. Then $d'(s) \geq 4$ and s has a neighbor in $V - (A \cup B)$. Now it follows from Proposition 2.1 that

$$d'(A \cap B) = d'(A - B) = 4 \text{ and } d'(A) = 5.$$

It is not difficult to check by a parity argument that all of these equalities cannot hold at the same time. This contradiction verifies (2). Finally, observe that (3) follows easily from (1) and (2). □

LEMMA 4.4. *If $d'(s) = 4$ and $|E(H)| \leq 2$, then there exists a feasible complete splitting of s, unless G' is of type* I, II, *or* III *in Figure* 4.1.

Proof. $d'(s) = 4$ implies that $|S| = 3$ or $|S| = 4$. If $|S| = 3$, then every admissible complete splitting of s will involve adding the edges uv and uw, where $S = \{u, v, w\}$

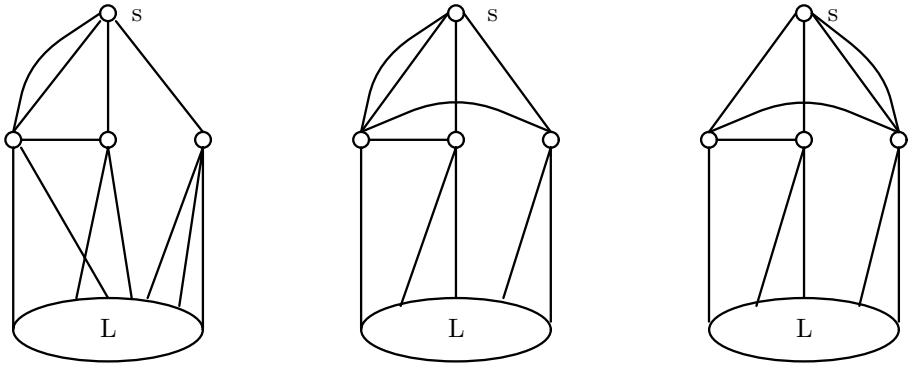


FIG. 4.1. *The exceptional cases when* $d'(s) = 4$ *and* $|S| = 3$. *For each of the three types, every nonempty subset of* L *has degree at least* 4.

and u is the vertex with two edges from s. Hence by Theorem 2.2 a feasible complete splitting exists if and only if $d_H(u) = 0$ and this is easily seen to be the case if and only if G' is not of type I, II, or III in Figure 4.1.

Consider the case $|S| = 4$. By Theorem 2.2, we may assume that $|E(H)| \geq 1$. Suppose first that $d_H(v) \leq 1$ for each $v \in H$ and let $uv \in E(H)$. Let w, z be the remaining vertices in S. It follows from Lemma 4.3 (2) that at least one of the pairs su, sw and su, sz is legal for splitting and, since the remaining pair of vertices in H are not adjacent, this leads to a feasible complete splitting of s.

Now assume that H contains the edges uv, uw. Let z be the fourth vertex of S. It follows from Lemma 4.3 that the pair su, sz is legal for splitting and hence, since v and w are not adjacent, there is a feasible complete splitting of s. □

LEMMA 4.5. *If* $d'(s) \geq 4$, *then there exists a legal pair unless* $d'(s) = 4$, $|S| = 3$, *and* G' *is of type* II *in Figure* 4.1.

Proof. If $d'(s) \geq 6$, then it follows easily from Lemma 4.3 that there exists a legal pair. Hence we may assume that $d'(s) = 4$. Suppose first that $|S| = 4$. It follows from Lemma 4.3 that if there is no legal pair, then every $x \in S$ has degree 3 in H. In this case $d_G(S, V - S) = 0$ follows, a contradiction.

If $|S| = 3$, $S = \{x, y, z\}$, we first note that if x is the vertex with two edges from s, then there is no dangerous set containing x and some other neighbor of s. Hence it follows that if there is no legal splitting involving x, then $xy, xz \in E(H)$. In this case G' is of type II in Figure 4.1. □

LEMMA 4.6. *If* $d'(s) = 2$, *then* $G' - s$ *can be made* 4*-edge-connected preserving simplicity by adding at most two edges.*

Proof. Let $D = G' - s$. If the two neighbors u, v of s are not adjacent, then uv makes D 4-edge-connected; thus we can assume that $uv \in E(D)$.

By our assumption either $d_D(u) = d_D(v) = 3$ or $d_D(u) = 4$ and $d_D(v) = 3$. The second case occurs if one of the two s-neighbors in G', say u, was previously connected to s by two parallel edges. Since $d'(s) = 2$, there are precisely two minimal critical sets in D. One of them contains u, the other contains v. It is easy to see that adding an arbitrary edge connecting these two minimal critical sets makes D 4-edge-connected. Let us consider the first case. If D contains a vertex a which is not adjacent to any of u, v, then by adding the edges ua, av we get a simple and 4-edge-connected graph.

If no such vertex exists, then $|V(D)| = 6$ and $D - \{u, v\} = K_4$ must hold. Now it is easy to see that adding any pair of edges joining u to a nonneighbor of u and v to a nonneighbor of v makes D 4-edge-connected. In the second case observe that the minimal critical set of D containing u has size at least 4 by Lemma 4.1 and hence it contains a vertex w which is not adjacent to v. Now adding the edge wv makes D 4-edge-connected. □

THEOREM 4.7 (see [23]). *For every 3-edge-connected simple graph $G = (V, E)$ on at least five vertices the following hold:*

(i) $OPT^4_S(G) \leq OPT^4_P(G) + 1$.

(ii) *If $OPT^4_P(G) \geq 3$, then $OPT^4_S(G) = OPT^4_P(G)$ unless $G = K_{3,3}$.*

Proof. Let $G' = (V + s, E \cup F)$ be the graph returned by Phase 2 of Frank's algorithm. If we are able to perform a sequence of legal splittings with the effect of adding a set of edges F' to G while preserving simplicity, then it follows from Theorem 2.2 and the fact that at the end of Phase 2 of Frank's algorithm $OPT^4_P(G) = d'(s)/2$, that the following hold:

$$OPT^4_P(G + F') = OPT^4_P(G) - |F'|, \tag{4.1}$$

$$OPT^4_S(G + F') \geq OPT^4_S(G) - |F'|. \tag{4.2}$$

Hence if we can show that $OPT^4_S(G+F') \leq OPT^4_P(G+F')+1$ (respectively, $OPT^4_S(G+F') = OPT^4_P(G+F')$), then it will follow that $OPT^4_S(G) \leq OPT^4_P(G)+1$ (respectively, $OPT^4_S(G) = OPT^4_P(G)$). We will use this observation several times below.

We first prove (i). By Lemma 4.2 we may assume that all minimal critical sets of G are singletons. By Lemma 4.6 we may also assume $d'(s) \geq 4$. Now we use Lemma 4.5 to perform legal splittings until we have $d'(s) = 4$ in the current G'. If at this point we have $|S| = 3$, then by the fact that $d'(S, V - S) \geq 4$, $|E(H)| \leq 2$ must hold and by Lemma 4.4, either there exists a feasible complete splitting or G' is of type I, II, or III in Figure 4.1. It is not difficult to see that in the latter case, too, we have $OPT^4_S(G' - s) = OPT^4_P(G' - s)$. So from this and the observation above we get that if $|S| = 3$ when $d'(s) = 4$, then $OPT^4_S(G) = OPT^4_P(G)$.

Hence we may assume that $|S| = 4$, and now it follows from Lemma 4.5 that we can still find one more legal splitting at this point. Then applying Lemma 4.6 we obtain that $OPT^4_P(G) + 1$ edges are sufficient. Thus in all of the cases we get that $OPT^4_S(G) \leq OPT^4_P(G) + 1$. This proves (i).

To prove (ii) suppose that $OPT^4_P(G) \geq 3$ and $G \neq K_{3,3}$ (for which we have $OPT^4_S(K_{3,3}) = OPT^4_P(K_{3,3}) + 1$). Note that $OPT^4_P(G) \geq 3$ implies $|V| \geq 6$. By Lemma 4.5 we can split off feasible pairs in G' until $d'(s) = 6$ holds. At this point we have $OPT^4_P(G' - s) = 3$. By our remark at the beginning of the proof, it follows that it is sufficient to prove that $OPT^4_S(G' - s) = 3$ holds for the current G'. We verify this by showing that $OPT^4_P(G) = 3$ implies $OPT^4_S(G) = 3$ for every $G \neq K_{3,3}$. In what follows let $G' = (V + s, E \cup F)$ be the graph returned by Phase 2 of Frank's algorithm applied to some simple graph G with $OPT^4_P(G) = 3$. Now $d'(s) = 6$ and hence $5 \leq |S| \leq 6$. By Lemma 4.2 we may assume that all minimal critical sets of G are singletons. If $|V| = 6$, the graph G is either the prism (that is, the complement of a cycle of length six) or a wheel, for which the desired equality trivially holds. In what follows we assume $|V| \geq 7$. Since G is 3-edge-connected and each neighbor of s has degree 3 in G, it follows that $|E(H)| \leq 7$ and, if $|S| = 5$, then $|E(H)| \leq 5$ (there must be at least four edges from S to $V - S$ in G').

Case 1. $|S| = 6$. We shall argue that we can always find a legal splitting with the property that after making this splitting at most two edges remain in the new graph H. By Lemma 4.4 this implies that there exists a feasible complete splitting of s and hence $OPT_S^4(G) = OPT_P^4(G)$.

If $|E(H)| = 7$, then H has two vertices u, v of degree 3 and, by Lemma 4.3, there is a legal splitting for the edge su (respectively, sv) with every edge sw where w is a nonneighbor of u (respectively, v). Hence we may assume that u and v are adjacent, because otherwise we can eliminate six edges of H by performing one legal splitting. Since G is 3-edge-connected, u and v cannot have two common neighbors x, y in H since then the degree of the set $X = \{u, v, x, y\}$ would be at most 2 in G. If u and v have no common neighbors, then it is easy to check that there is a legal splitting (involving one of the edges su, sv) such that at most two edges remain in H afterwards, because su (sv) can be split off with the edge from s to each of the two neighbors of v (u). So we can assume that u and v have precisely one neighbor x in common. Now it is easy to check, using Lemma 4.3, that we can always find a legal splitting that eliminates at least five edges from H.

If $|E(H)| = 6$, then either each of the vertices in H has degree 2 and it follows from Lemma 4.3 (1) that the desired legal splitting exists, or there is a vertex u of degree 3 in H. Since G is 3-edge-connected, we cannot have all six edges of H inside the graph induced by u and its neighbors and hence again, by Lemma 4.3, the desired legal splitting exists.

If $|E(H)| = 5$, $|E(H)| = 4$, or $|E(H)| = 3$ and H has a vertex of degree at least 2, then it is easy to see that we can find a legal splitting eliminating all but two edges in H. Finally, if $|E(H)| = 3$ and each vertex has degree 1 in H, then every legal splitting has the desired property.

Case 2. $|S| = 5$. Since $d'(s) = 6$ it follows that we have two parallel edges between s and some vertex $v \in S$. Let $S = \{v, x, y, z, w\}$. Since we started from a 3-edge-connected graph G, the vertices $\{x, y, z, w\}$ (all of which, by our assumption, have degree 3 in G) do not induce a K_4. Hence it follows from Lemma 4.3 that there is a legal splitting which involves two vertices in the set $\{x, y, z, w\}$, say z, w. Now the remaining neighbors of s, $\{v, x, y\}$ induce a graph with at most two edges, since G' is 4-edge-connected and each of the vertices in S has degree 3 in G. Now, by Lemma 4.4, either there is a feasible complete splitting of s in the current G', or G' is of type I, II, or III in Figure 4.1 in which case, as we remarked in the proof of (i), $OPT_S^4(G' - s) = OPT_P^4(G' - s)$. Hence by the remark at the beginning of the proof, we have shown that $OPT_S^4(G) = OPT_P^4(G)$. □

It can be verified—by analyzing the algorithm of [22], say—that any simple graph G has an optimal simple 3-edge-connected augmentation G' for which $OPT_P^4(G) = OPT_P^4(G') + |E(G')| - |E(G)|$. This and the results of this section show that $g(4) = 1$.

We conjecture that if the starting graph is already $(k-1)$-edge-connected, the bound $3k^4/2$ in Theorem 3.3 can be replaced by a linear function of k. Perhaps k or $k+1$ is sufficient.

5. Nonuniform demands. The augmentation problem without the simplicity-preserving requirement is solvable in polynomial time even if the target connectivity is not uniform but is given by a symmetric function $r : V \times V \to Z_+$ on pairs of vertices of the starting graph $G = (V, E)$ (and the goal is to find a smallest set F of new edges such that in $G' = (V, E \cup F)$ the local edge-connectivity $\lambda(u, v)$ is at least $r(u, v)$ for any pair (u, v) of vertices). This more general version was solved by Frank [8].

For every subset $\emptyset \neq X \subset V$ let us define

$$R(X) = \max\{r(u,v) : u \in X, v \in V - X\}.$$

Let $k = \max\{r(u,v) : (u,v) \in V \times V\}$. (For simplicity, to avoid the so-called marginal components [8], we assume that $r(u,v) \geq 2$ for each pair $u, v \in V$.) By Menger's theorem the extended graph (or the augmented graph) satisfies the requirements if and only if

$$d'(X) \geq R(X) \text{ for every } \emptyset \neq X \subset V. \tag{5.1}$$

As in the uniform-demand case, Frank used the same splitting off method in his proof. In fact, the algorithm he gave is identical to the algorithm of the uniform case except, that after adding k parallel edges between a new vertex s and each vertex of V, during the deletion part in Phase 1 we must maintain (5.1) instead of (1.1) and during the splitting off in Phase 3 the local edge-connectivities must be preserved everywhere between pairs of vertices in V. (For more details we refer to [8].) The corresponding Phase 3 can be done by the following result of Mader.

We say that two edges st and su form an *admissible pair* in $H = (V + s, E')$ if after splitting off st and su the local edge-connectivities remain the same between vertices of V. (Note that splitting off never increases the local edge-connectivity.)

THEOREM 5.1 (see [18]). *Let $H = (V + s, E')$ be a connected undirected graph with $d(s) \neq 3$ for which there is no cut-edge incident to s. Then there is an admissible pair st, su of edges.*

In this section our goal is to prove the counterparts of Theorems 3.1, 3.3, and 3.4 by showing that in the case of the simplicity-preserving version of the nonuniform augmentation problem, there exist similar functions $f'(k)$ and $g'(k)$ such as $f(k)$ and $g(k)$ in the uniform case and the problem is solvable in polynomial time if k is fixed.

We shall use a similar approach that we used for the uniform case. In fact, the crucial part of the generalization is to prove that a similar statement (Lemma 5.4 below) corresponding to Theorem 2.2(b) holds in this case, too. This will ensure that the number of edges which are nonadmissible with respect to some fixed edge st can be bounded by a function of k. However, to prove this we must modify the concept of admissibility and also Phase 3 of Frank's algorithm (in the nonuniform case). The reason is that although the goal is to satisfy (5.1) only, the solution given by Frank's algorithm will guarantee more and it will maintain the local edge-connectivities of the extended graph constructed in Phase 2. On the other hand, by this extra property of the solution it is easy to see that certain optimal augmentations cannot be obtained by Frank's algorithm. For example, let G be the disjoint union of two stars $K_{1,m}$ and let $r(u,v) = 2$ for each pair u, v of vertices. Then the optimal solution produced by Frank's algorithm must be a set of m independent edges between the two stars — although any set of m independent edges on the leaves would be a good augmenting set provided that there are at least two edges connecting the two stars.

Moreover, in the simplicity-preserving version sometimes the minimum number of new edges to be added to satisfy the requirements after the first two phases of Frank's algorithm — given by the local edge-connectivities in G' — is more than the optimum value with respect to the original demands given by the function r. For example, the first two phases of Frank's algorithm applied to the graph in Figure 5.1 add the edges sx, sy, sv, sz. Then although there exists a proper augmenting set of size two (xy and vz), there is no legal splitting in Phase 3, since $\lambda'(a,b) = 6$ must also be maintained.

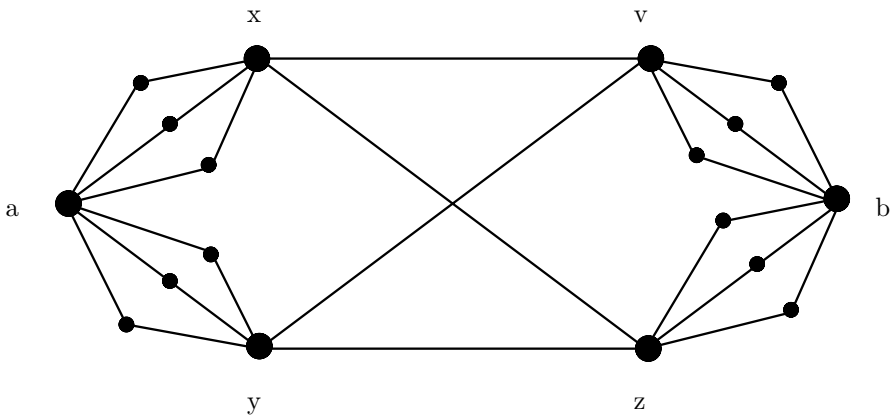


FIG. 5.1. $r(x,y) = r(v,z) = 6$ *and* $r \equiv 2$ *otherwise.*

Therefore if one wants to preserve simplicity as well, it is better to work with the following less restrictive definition of admissibility in Phase 3.

Let $G = (V + s, E')$ be a graph for which (5.1) holds, where the function R is defined by the nonuniform requirements, as before. Then we say that two edges st and su form an *R-admissible pair* (or an admissible pair, if R is clear from the context) if splitting off st and su does not violate (5.1). A set $\emptyset \neq X \subset V$ is *tight* if $d(X) = R(X)$ holds. If $d(X) \leq R(X) + 1$, we say that X is *dangerous*. Let $s(X) := d(X) - R(X)$. It is easy to see that st, su is an admissible pair if and only if there is no dangerous set X with $t, u \in X$. Now we briefly list some results we shall use from [7] and [8].

PROPOSITION 5.2 (see [8]). *For every pair $X, Y \subseteq V$ at least one of the following inequalities holds:*

$$R(X) + R(Y) \leq R(X \cap Y) + R(X \cup Y), \tag{5.2a}$$

$$R(X) + R(Y) \leq R(X - Y) + R(Y - X). \tag{5.2b}$$

It is easy to check that the proofs of [7, Claims 4.2, 3.2, 4.3] work the same way if we use our definition of admissibility. Thus we can obtain three similar statements. The first one gives that if X is a dangerous set (with respect to some edge st), then $d(s, X) \leq d(s, V - X)$. To state the lemma corresponding to [7, Claim 3.2] we need one more definition. (The third claim we get will be mentioned in the proof below.) The *contraction* of a subset X of vertices in a graph $G = (V, E)$ means that we delete X and replace it by a new vertex v_X and then add $d_G(v, X)$ new parallel edges between each $v \in V - X$ and v_X. The resulting graph is denoted by G/X. After the contraction of some subset X, we define the new requirement function r' in G/X as expected: $r'(v_X, w) = \max\{r(x, w) : x \in X\}$ and $r'(u, w) = r(u, w)$ for $u, w \in V - X$. The corresponding function R' on the subsets of $V(G/X)$ is defined by r'.

LEMMA 5.3. *Let T be a tight set. A pair st, su of edges is R-admissible in G if the corresponding pair of edges is R'-admissible in G/T.*

Now we are ready to prove the lemma we need.

LEMMA 5.4. *Suppose that $G = (V + s, E')$ is 2-edge-connected. Then for every edge st the number of edges su for which the pair st, su is nonadmissible is at most $2k^2 - 2k$.*

Proof. Let S denote the set of neighbors of s and let $W \subset S$ denote those neighbors u for which st and su form a nonadmissible pair. By one of our previous remarks this means that there exists a family $\mathcal{F}$ of dangerous sets, each containing t, which covers every vertex of W. Let us fix such a family for which $|\mathcal{F}|$ is minimal and subject to this $\sum_{X\in\mathcal{F}}|X|$ is maximal. We shall prove that $|\mathcal{F}| \leq 2k-2$ holds. (Note that by [7, Claim 4.3], two dangerous sets cannot cover the whole S.)

First we claim that every dangerous set induces a connected subgraph. To see this, assume that X is dangerous and has two components X_1 and X_2 such that, without loss of generality, $R(X)$ is attained on a pair $u \in X_1$, $v \in V - X$. Then $R(X) \leq R(X_1) \leq d(X_1) = d(X) - d(X_2) \leq d(X) - 2 \leq R(X) - 1$, since $d(X_2) \geq 2$ by 2-edge-connectivity, a contradiction.

Let us consider a tight set M_1 and a dangerous set M_2, both containing t. For these two sets (5.2b) cannot hold since otherwise by Proposition 2.1 we have

$$0 + 1 \geq s(M_1) + s(M_2) \geq s(M_1 - M_2) + s(M_2 - M_1) \\ + 2d(M_1 \cap M_2, (V + s) - (M_1 \cup M_2)) \geq 0 + 0 + 2,$$

a contradiction. (We used that $d(M_1 \cap M_2, (V+s)-(M_1 \cup M_2)) \geq 1$ by the existence of the edge st.) Thus (5.2a) must hold by Proposition 5.2, which implies that $s(M_1 \cap M_2)+s(M_1 \cup M_2) \leq s(M_2)$. From this it follows that if M_2 is also tight, then $M_1 \cap M_2$ and $M_1 \cup M_2$ are both tight (as we remarked, $M_1 \cup M_2 \neq V$). Another consequence is that if $M_2 \in \mathcal{F}$, then $M_1 \subseteq M_2$ must hold, otherwise $2 \leq s(M_1 \cup M_2) \leq s(M_2) \leq 1$ would follow by the choice of $\mathcal{F}$. These observations imply that if there exists a tight set which contains t, then there exists a unique maximal tight set M containing t. Moreover, if such a tight set M exists, then M is a subset of every member of $\mathcal{F}$ and $d(s, M) \leq d(M) = R(M) \leq k$ holds.

By Lemma 5.3 the contraction of a tight set does not decrease the number of edges which are nonadmissible with respect to st. Thus in the rest of the proof we shall assume that every tight set is a singleton.

We say that a pair X, Y of members of $\mathcal{F}$ is an *a-pair* if (5.2a) holds for X and Y. Otherwise the pair is a *b-pair*. If $X, Y \in \mathcal{F}$ is an a-pair, we get

$$1 + 1 \geq s(X) + s(Y) \geq s(X \cap Y) + s(X \cup Y) \geq 0 + 2,$$

since $s(X \cup Y) \geq 2$ by the choice of $\mathcal{F}$. This shows that their intersection $X \cap Y$ must be tight. Therefore $X \cap Y = M = \{t\}$ holds for each a-pair X, Y.

Suppose now that X, Y is a b-pair and Y, Z is an a-pair. We claim that then X, Z must be a b-pair. To see this, suppose X, Z is an a-pair. Then $Z - M$ is disjoint from $(X \cup Y) - M$. Furthermore, there is precisely one edge — the edge st — from $X \cap Y$ to $(V + s) - (X \cup Y)$ by the inequality

$$1 + 1 \geq s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2d(X \cap Y, (V + s) - (X \cup Y)).$$

From this it follows that there is no edge between M and $Z - M$; hence Z is not connected, contradicting the fact that every dangerous set induces a connected graph.

Hence $\mathcal{F}$ can be partitioned into subfamilies $\mathcal{F}_1, ..., \mathcal{F}_r$ such that X, Y is a b-pair if and only if X and Y are in different subfamilies. Suppose that one of these subfamilies $\mathcal{F}_i$ has size at least k. Since each member in this subfamily is connected and $(X-M)\cap(Y-M) = \emptyset$ for different members $X, Y \in \mathcal{F}_i$, M has at least k different neighbors in V. Since M is connected to s as well, $k \geq R(M) = d(M) \geq k+1$ follows, a contradiction. Thus each subfamily in the partition has size at most $k - 1$. This

implies that if $|\mathcal{F}| \geq 2k-1$, then there are three sets $X, Y, Z \in \mathcal{F}$ such that they are pairwise b-pairs. Since for a b-pair X, Y the sets $X - Y$ and $Y - X$ are both tight, and hence singletons, the minimality of $\mathcal{F}$ implies that there exists a set N such that $X \cap Y = Y \cap Z = X \cap Z = N$ holds. Using (5.2b), this gives that the only edge that leaves N in G is st, a contradiction.

Hence we obtain $|\mathcal{F}| \leq 2k-2$. Let $\alpha := d(s,t) \leq k$. Now for the number β of edges which are nonadmissible with respect to st we obtain $\beta \leq (2k-2)(k+1-\alpha)+\alpha-1 \leq 2k^2-(2k-3)\alpha-3 \leq 2k^2-2k$, as required. □

From Lemma 5.4, following the proof ideas of Theorems 3.1, 3.3, and 3.4 one can obtain the corresponding results for the nonuniform version, that is, the existence of the polynomials f', g' below. The details, which are similar, are omitted, but some remarks must be added. First of all, we modify Frank's nonuniform algorithm in such a way that (except the last part, where only a small number of edges are present) in Phase 3 we split off edges which form admissible pairs in the sense we defined. This is necessary, since otherwise Lemma 5.4 is not valid (see the example of two stars at the beginning of this section). Efficient splitting off algorithms preserving local edge-connectivities are in [8], [12]. They can be modified to work with our admissibility conditions. Furthermore, the points where we need more involved arguments are the extension of Proposition 3.5 and the existence of simple graphs with maximum degree $k = \max\{r(u,v) : u, v \in V\}$ satisfying the nonuniform demands. (In the latter case a result from [10] can be referred to.) Also note that our assumption $r \geq 2$ implies that the 2-edge-connectivity condition in Lemma 5.4 is fulfilled.

THEOREM 5.5. *Let $G' = (V+s, E \cup F)$ be a graph at the end of the second phase of Frank's (nonuniform) algorithm such that*

$$d'(s) \geq f'(k). \tag{5.3}$$

Then there exists a feasible complete splitting from s, where feasibility is meant with respect to (5.1).

THEOREM 5.6. *If $OPT_P^k(G) \geq f'(k)/2$, then $OPT_S^k(G) = OPT_P^k(G)$ holds.*

THEOREM 5.7. *$OPT_S^k(G) \leq OPT_P^k(G) + g'(k)$ for any G and nonuniform demands $r(u,v)$, where $k = \max\{r(u,v) : u, v \in V\}$.*

So far we have no proof for a counterpart of Theorem 3.6. This would improve the efficiency of the algorithm such as Theorem 3.6 did in the uniform case.

6. Remarks. In this last section some remarks are made related to possible extensions of the simplicity-preserving edge-connectivity augmentation problem.

The "subset-problem," where the goal is to find a simplicity-preserving augmentation which makes a subset $X \subset V$ k-edge-connected, was mentioned as the next open problem to be studied (at least for $k = 3$) in [23]. Observe that this is a special case of the nonuniform demand version. (On the other hand, the subset-problem can be solved by just slightly modifying our proofs of the uniform case as well.)

For the directed version of our problem a similar function $f(k)$, such as in Theorem 3.1, does not exist. We have found a family G_i^k of digraphs for every $k \geq 1$ with $i \leq OPT_P^k(G_i^k) < OPT_S^k(G_i^k)$, $i = 1, 2, \ldots$.

Although the weighted version of the edge-connectivity augmentation problem is NP-hard, the special case where the weight function is induced by weights given on the vertices (and the weight of a new edge is the sum of the weights of its end-vertices) is polynomially solvable [8]. Our arguments do not apply for arbitrary weight functions of this type in the simplicity-preserving augmentation problem. The reason is the small modification we made in Phase 2 of the algorithm. For example, consider a star

$K_{1,m}$, where m is odd, $k = 2$, and the weights on the vertices are uniformly 2, except the center v of the star, whose weight is 1.

Some of the above results are valid in the more general case where the starting graph G is not simple, but the augmenting set F must not contain parallel edges and edges which are parallel to edges of G. These details are omitted.

Finally we remark that the complexity of the version where the augmenting set F must not contain parallel edges, but a new edge may be parallel to an edge of the starting graph, is still open.

REFERENCES

[1] J. Bang-Jensen, H. N. Gabow, T. Jordán, and Z. Szigeti, *Edge-connectivity augmentation with partition constraints*, in Proc. 9th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA), SIAM, Philadelphia, PA, 1998, pp. 306–315.

[2] J. Bang-Jensen and T. Jordán, *Edge-connectivity augmentation preserving simplicity*, in Proc. 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Los Alamitos, CA, pp. 486–495.

[3] A. A. Benczúr, *Parallel and fast sequential algorithms for undirected edge augmentation*, Math. Programming Ser. B, to appear.

[4] G.-R. Cai and Y.-G. Sun, *The minimum augmentation of any graph to a K-edge-connected graph*, Networks, 19 (1989), pp. 151–172.

[5] E. Cheng and T. Jordán, *Successive edge-connectivity augmentation problems*, Math. Programming, Ser. B, to appear.

[6] K. Eswaran and R. E. Tarjan, *Augmentation Problems*, SIAM J. Comput., 5 (1976) pp. 653–665.

[7] A. Frank, *On a theorem of Mader*, Discrete Math., 101 (1992), pp. 49–57.

[8] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.

[9] A. Frank, *Connectivity augmentation problems in network design*, in Mathematical Programming: State of the Art, J. R. Birge and K. G. Murty, eds., The University of Michigan, Ann Arbor, MI, 1994, pp. 34–63.

[10] H. Frank and W. Chou, *Connectivity considerations in the design of survivable networks*, IEEE Trans. Circuit Theory, 17 (1970), pp. 486–490.

[11] H. N. Gabow, *Applications of a poset representation for edge-connectivity and graph rigidity*, in Proc. 32nd IEEE Symposium on Foundations of Computer Science, San Juan, Puerto Rico, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 812–821.

[12] H. N. Gabow, *Efficient splitting off algorithms for graphs*, in Proc. 26th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1994, pp. 696–705.

[13] T-S. Hsu and M-Y. Kao, *Optimal augmentation for bipartite componentwise biconnectivity in linear time*, in Algorithms and Computation (Proc. ISAAC '96), Lecture Notes in Comput. Sci. 1178, 1996, Springer-Verlag, New York, pp. 213–222.

[14] B. Jackson, *Some remarks on arc-connectivity, vertex splitting, and orientation in graphs and digraphs*, J. Graph Theory, 12 (1988), pp. 429–436.

[15] T. Jordán, *Two NP-complete Augmentation Problems*, IMADA, Odense University, Denmark, Preprint No. 8, 1997.

[16] G. Kant and H. L. Bodlaender, *Planar graph augmentation problems*, in Algorithms and Data Structures (Proc. WADS '91), Lecture Notes in Comput. Sci. 519, Springer-Verlag, New York, 1991, pp. 286–298.

[17] L. Lovász, *Combinatorial Problems and Exercises*, North–Holland, Amsterdam, 1979.

[18] W. Mader, *A Reduction Method for Edge-Connectivity in Graphs*, Ann. Discrete Math., 3 (1978), pp. 145–164.

[19] H. Nagamochi and P. Eades, *Edge-splitting and edge-connectivity augmentation in planar graphs*, in Proc. 6th Conf. on Integer Programming and Combinatorial Optimization (IPCO), Rice University, Houston, TX, 1998, to appear.

[20] H. Nagamochi and T. Ibaraki, *Computing edge-connectivity in multigraphs and capacitated graphs*, SIAM J. Discrete Math. 5, (1992), pp. 54–66.

[21] H. Nagamochi and T. Ibaraki, *Deterministic $\tilde{O}(nm)$ time edge-splitting in undirected graphs*, J. Combin. Optim., 1 (1997), pp. 5–46.

[22] D. Naor, D. Gusfield, and C. Martel, *A fast algorithm for optimally increasing the edge-connectivity*, SIAM J. Comput., 26 (1997), pp. 1139–1165.
[23] S. Taoka, D. Takafuji, and T. Watanabe, *Simplicity preserving augmentation of the edge-connectivity of a graph*, TR IEICE, 1994, pp. 49–56.
[24] T. Watanabe and M. Yamakado, *A linear time algorithm for smallest augmentation to* 3*-edge-connect a graph*, IEICE Trans. Fundamentals, E76-A (1993), pp. 518–530.
[25] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, J. Comput. System Sci., 35 (1987), pp. 96–144.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 624–632, November 1998

007

DIMENSION OF PROJECTIONS IN BOOLEAN FUNCTIONS*

RAMAMOHAN PATURI AND FRANCIS ZANE†

Abstract. A projection is a subset of $\{0,1\}^n$ given by equations of the form $x_i = x_j$, $x_i = \bar{x}_j$, $x_i = 0$, and $x_i = 1$, where for $1 \le i \le n$, x_i are Boolean variables and $\bar{x}_i$ are their complements. We study monochromatic projections in 2-colorings of an n-dimensional Boolean cube. We also study the dimension of the largest projection contained in a set specified by its density. We prove almost matching lower and upper bounds on the density of a set required to guarantee the existence of a d-dimensional projection. We also prove almost tight upper and lower bounds on the dimension of monochromatic projections in arbitrary Boolean functions. We then prove almost tight upper and lower bounds on the dimension of monochromatic projections in Boolean functions represented by low degree GF(2) polynomials. It follows from these lower bounds that low-degree GF(2) polynomials can define Boolean functions which are close to being extremal with respect to the property of having no large dimensional monochromatic projections.

Key words. projections, Ramsey theory, circuit complexity

AMS subject classifications. 05D10, 68Q15, 68R05

PII. S0895480197318313

1. Introduction.

In this paper, we study monochromatic projections in 2-colorings of an n-dimensional Boolean cube. We also consider the related density question: what is the density required to guarantee the existence of a d-dimensional projection? A projection is a subset of $\{0,1\}^n$ given by equations of the form $x_i = x_j$, $x_i = \bar{x}_j$, $x_i = 0$, and $x_i = 1$, where for $1 \le i \le n$, x_i are the Boolean variables and $\bar{x}_i$ are their complements. Thus, a projection is an affine subspace of the n-dimensional $GF(2)$ space $\{0,1\}^n$. The dimension of a projection is its dimension as an affine subspace. A projection P is monochromatic under a Boolean function f if $P \subseteq f^{-1}(1)$ or $P \subseteq f^{-1}(0)$. Projections are closely related to Boolean algebras, whose Ramsey-theoretic properties have been studied extensively [4]. Boolean algebras are projections where equations of the form $x_i = \bar{x}_j$ are not allowed, so the sets are always oriented in a canonical direction. Gunderson, Rödl, and Sidorenko [3] recently obtained almost matching bounds on the density required for the existence of a d-dimensional Boolean algebra. They also obtained almost tight bounds for the dimension of the largest monochromatic Boolean algebras under colorings of the Boolean cube.

In addition to being natural generalizations of Boolean algebras, projections are relevant to the study of circuit complexity of Boolean functions. For example, it is shown in [5] that if the set of satisfying solutions of a 2-CNF (conjunctive normal form with two literals per clause) is large, then it must contain a large dimensional projection. The existence of such *nice* subsets gives one a handle to construct *hard* functions for a given class of Boolean circuits. In particular, Boolean functions which do not have large dimensional monochromatic projections require large size depth-3 unbounded fan-in Boolean circuits with bottom fan-in 2. An interesting open question is whether Boolean functions computable by linear size circuits have $\omega(n^{3/4} \log n)$-dimensional monochromatic projections. A positive answer to this question implies that certain explicitly defined Boolean functions in NC (the class of Boolean functions

*Received by the editors March 12, 1997; accepted for publication (in revised form) December 15, 1997; published electronically September 1, 1998.
http://www.siam.org/journals/sidma/11-4/31831.html
†Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093 (paturi@cs.ucsd.edu, francis@cs.ucsd.edu).

computable by logspace uniform circuits of polylogarithmic depth and polynomial size) require nonlinear circuit size [5]. Proving lower bounds on the circuit size of interesting explicit Boolean functions is a fundamental challenge in complexity theory.

In this paper, we obtain density results for projections using techniques similar to those employed in [3]. We show that the existence of projections requires much lower density than in the case of Boolean algebras. We also obtain bounds on the dimension of the largest monochromatic projections in arbitrary Boolean functions. In addition, we consider the question of constructing Boolean functions which do not have large monochromatic projections. Although we do not have any explicit constructions, we show that there are functions much simpler than arbitrary Boolean functions which have this property. More precisely, we show that there exist degree-q GF(2) polynomials representing Boolean functions which have only $O(q(n \log n)^{1/q})$-dimensional monochromatic projections. It follows that there are Boolean functions represented by logarithmic degree polynomials which are nearly extremal with respect to the size of monochromatic projections. Pudlák, Rödl, and Savichý [6] and Razborov [7] obtained similar "low complexity" probabilistic constructions for combinatorial objects. We also show that the bounds obtained by the probabilistic technique are almost tight: given a degree-q GF(2) polynomial, we show how to construct a $\Omega(qn^{1/q})$-dimensional monochromatic projection.

We first introduce some definitions and state our results precisely. The following sections present the proofs of our results.

1.1. Definitions. Let $[n] = \{1, 2, \ldots, n\}$. Let $B_n = \{0,1\}^n$ denote the n-dimensional Boolean cube. We will also regard B_n as an n-dimensional $GF(2)$ vector space. A projection $P \subseteq B_n$ is the set of all $(x_1, x_2, \ldots, x_n) \in B_n$ satisfying a system of equations of the form $x_i = x_j$, $x_i = \bar{x}_j$, $x_i = 0$, and $x_i = 1$. x_i are Boolean variables and $\bar{x}_i$ is the complement of x_i. The dimension of a projection is its dimension as an affine subspace. It is convenient to think of a d-dimensional projection as a partition $\{A_0, B_0, A_1, B_1, \ldots, A_d, B_d\}$ of $\{1, 2, \ldots, n\}$ with $A_0, B_0, B_1, \ldots, B_d$ possibly empty. A_0 and B_0 are the sets of variables which are set to 0 and 1, respectively. For $1 \leq i \leq d$, A_i is nonempty and all its variables are equal to each other. Furthermore, the variables in B_i are equal to each other and equal to the complement of the variables in A_i. We obtain a set of free variables of a projection by selecting a representative from each class A_i for $1 \leq i \leq d$.

We also need some notation to deal with hypergraphs. A d-uniform hypergraph is a pair $G = (V, E)$ with vertex set V and hyperedge set $E \subseteq \binom{V}{d}$, where $\binom{V}{d}$ is the set of all d-subsets of V. A d-partite d-uniform hypergraph is a $d+1$-tuple $G = (X_1, X_2, \ldots, X_d, E)$, where X_i are pairwise disjoint sets and $(\cup_{i=1}^{d} X_i, E)$ is a d-uniform hypergraph whose edges have exactly one point from each X_i. The sets X_i are called partite sets. The complete d-partite d-uniform hypergraph with two vertices in each partite set and having 2^d edges is denoted by $K^{(d)}(2, 2, \ldots, 2)$. Let $ex(n, H)$ be the maximum number of d-hyperedges in a hypergraph on n vertices which does not contain a copy of H.

1.2. Statement of the results. Let $\rho(n, d)$ denote the maximum density of a subset $A \subseteq B_n$ which does not contain a d-dimensional projection. Our first result gives upper and lower bounds on $\rho(n, d)$.

THEOREM 1.

$$2^{-\frac{n \log(2d+2)}{2^d} - 2} \leq \rho(n, d) \leq 2^{-\frac{n}{d(2^d-1)} + 1}.$$

Let f be a Boolean function with the domain $\{0,1\}^n$. Let $\tau(f)$ be the dimension of the largest monochromatic projection of f. Let $d_n = \min_f \tau(f)$.

THEOREM 2.

$$\log n - \log\log n + o(1) \le d_n \le \log n + \log\log\log n + o(1).$$

Compared to the more restricted class of Boolean algebras, the dimension of the largest projection which exists in an arbitrary subset of B_n is much larger. Let $\rho^{BA}(n,d)$ and d_n^{BA} be the analogues of $\rho(n,d)$ and d_n defined for Boolean algebras rather than projections. The corresponding results for these quantities from [3] are

$$c_1(d)n^{-\frac{d}{2^{d+1}-2}(1-o(1))} \le \rho^{BA}(n,d) \le c_2(d)n^{-\frac{1}{2^d}}$$

and

$$\log\log n - (1+o(1))\log\log\log n \le d_n^{BA} \le \log\log n + \log\log\log n.$$

We next consider the problem of constructing objects that match the bounds established earlier. We show that the set of codewords of a "good" code (that is, a code with constant rate and linear distance) can only contain bounded dimensional projections. However, our technique involving codes does not help us in constructing sets of size $c2^n$ whose largest projection has dimension $O(\log n)$. Furthermore, it is not clear how to use codes to construct Boolean functions with no large monochromatic projections, since the set of noncodewords may contain large projections.

Although we could not construct such Boolean functions, our next result shows that simple functions exist which do not contain monochromatic projections of dimension larger than $\log n + \log\log\log n$. We use low-degree GF(2) polynomials to represent Boolean functions. Using a probabilistic argument, we obtain the following.

COROLLARY 1. *There are degree q GF(2) polynomials whose largest monochromatic projection has dimension $O(q(n\log n)^{1/q})$.*

At the extreme end, we have the following.

COROLLARY 2. *There exists a GF(2) polynomial of degree $\lceil \log n + \log\log\log n + o(1)\rceil$ which has no monochromatic projections of dimension greater than $\lceil \log n + \log\log\log n + o(1)\rceil$.*

We also prove that this bound is almost tight.

COROLLARY 3. *Every degree q GF(2) polynomial contains a monochromatic projection of dimension $\Omega(qn^{1/q})$.*

Open problems.

1. Construct a Boolean function with the largest monochromatic projection of dimension $O(\log n)$.
2. Construct degree q GF(2) polynomials with the largest monochromatic projection of dimension $O(q(n\log n)^{1/q})$.

2. Dimension of projections in arbitrary Boolean functions. We adopt the techniques of Erdős [2] and Gunderson, Rödl, and Sidorenko [3] to obtain upper and lower bounds on the density required to guarantee the existence of a d-dimensional projection.

LEMMA 2.1. *For r-uniform, r-partite hypergraphs with n/r nodes in each part,*

$$ex(n, K^{(r)}(2,\ldots,2)) \le 2(n/r)^{r-\frac{1}{2^{r-1}}}$$

when $n \ge 16r$.

Proof. Let $G = (X_1, X_2, \ldots, X_r, E)$. The proof is by induction on r.

If $r = 2$, the lemma says that every $(\frac{n}{2}, \frac{n}{2})$-node bipartite graph (X_1, X_2, E) with $|E| \geq 2(\frac{n}{2})^{\frac{3}{2}}$ has a 4-cycle. The number of pairs of nodes (counted with repetition) in X_2 with a common neighbor in X_1 is $m = \sum_{v \in X_1} \binom{\deg(v)}{2}$. If this is larger than $\binom{n/2}{2}$, the number of distinct pairs of nodes in X_2, then some pair is counted twice, and a 4-cycle exists. m is minimized when every node in X_1 has average degree $\frac{|E|}{(n/2)} > 2\sqrt{\frac{n}{2}}$. In this case, $m = \frac{n}{2}\binom{2\sqrt{n/2}}{2} > \binom{n/2}{2}$ for $n \geq 0$.

If $r > 2$, by hypothesis $|E| \geq 2(\frac{n}{r})^{r - \frac{1}{2^{r-1}}}$. We first find two nodes $a_1, a_2 \in X_r$ for which there are many pairs of hyperedges $(y_1, \ldots, y_{r-1}, a_1)$ and $(y_1, \ldots, y_{r-1}, a_2)$. We then select those two nodes for part r of the r-partite graph $K^{(d)}(2, 2, \ldots, 2)$, and use induction to select the remaining parts. To show that such a pair of nodes exist, we use the following lemma.

LEMMA 2.2 (Erdős). *Let S be a set of N elements, $y_1, \ldots, y_N$, and let $A_i, 1 \leq i \leq k$, be subsets of S. Let w be such that $\sum_i |A_i| \geq \frac{kN}{w}$. If $k \geq 2l^2 w^l$, then there exist $A_{i_1}, \ldots, A_{i_l}$ such that $|\cap_j A_{i_j}| \geq \frac{N}{2w^l}$.*

Now, apply the lemma with S as the set of all $(r-1)$-tuples from $X_1 \times \cdots \times X_{r-1}$ and A_i as the set of all such tuples which, when extended by the ith element of X_r, belong to E. Then $N = (\frac{n}{r})^{r-1}, k = n/r, w = \frac{1}{2}(\frac{n}{r})^{\frac{1}{2^{r-1}}}$, and $l = 2$. Since $n \geq 16r$, the condition on k in the lemma is satisfied. There exist a_1, a_2 which have

$$\frac{N}{2w^2} \geq \frac{(\frac{n}{r})^{r-1}}{2(\frac{1}{2}(\frac{n}{r})^{\frac{1}{2^{r-1}}})^2} \geq 2\left(\frac{n}{r}\right)^{(r-1)-\frac{1}{2^{r-2}}}$$

$(r-1)$-tuples in common. By induction, there exists a $K^{(r-1)}(2, \ldots, 2)$ among these $(r-1)$-tuples. This can then be extended to a $K^{(r)}(2, \ldots, 2)$ by extending each such $(r-1)$-tuple $(y_1, \ldots, y_{r-1})$ to $(y_1, \ldots, y_{r-1}, a_1)$ and $(y_1, \ldots, y_{r-1}, a_2)$. ∎

We now apply the lemma to obtain an upper bound on $\rho(n, d)$, the density required to guarantee the existence of a d-dimensional projection.

LEMMA 2.3. *For $n \geq 4$,*

$$\rho(n, d) \leq 2^{-\frac{n}{d(2^d-1)}+1}.$$

Proof. We only consider the case where $d \leq \log n$ since, otherwise, the theorem is vacuously true.

Given a set A with $|A| \geq 2^{n(1-\frac{1}{d(2^d-1)})+1}$, one can obtain a d-dimensional projection as follows.

Partition $[n]$ into d classes, X_i, such that the largest class size is $\lceil n/d \rceil$ and the smallest class size is $\lfloor n/d \rfloor$. Consider the following d-uniform, d-partite hypergraph H. The ith part H_i will have $2^{|X_i|}$ vertices, indexed by $\{0,1\}^{|X_i|}$. For each point $a \in A$, include the hyperedge $(a|_{X_1}, \ldots, a|_{X_d})$ obtained by restricting a to the set of coordinates which appear in X_i.

Since this mapping between points and hyperedges is bijective, H has $2^{n(1-\frac{1}{d(2^d-1)})+1}$ hyperedges, and by the construction, H is d-uniform, d-partite with $d2^{n/d}$ vertices. By the preceding lemma, and by the hypothesis $n \geq 4$, it follows that H must contain a $K^{(d)}(2, 2, \ldots, 2)$, which we denote by $G = (Y_1, Y_2, \ldots, Y_d, E')$.

Given any two points of $\{0,1\}^m$ for $m \geq 1$, we can obtain a one-dimensional projection by the following: for each coordinate, if both points have the same value in that coordinate, set the corresponding variable to that constant value. Since the

points are distinct, there is at least one coordinate which is 0 on one point and 1 on the other. Fix one such coordinate and a variable to denote the value in that position. All variables that were not set to constants are equated to this variable or its negation so that the collection of equations determines precisely the two points.

It is now clear that the Cartesian product $Y_1 \times Y_2 \times \cdots \times Y_d$ is a d-dimensional projection. □

LEMMA 2.4.

$$\rho(n,d) \geq 2^{-\frac{n\log(2d+2)}{2^d}-2}$$

for all sufficiently large n.

Proof. Define $\varepsilon = 2^{-\frac{n\log(2d+2)}{2^d}-2}$. Consider the random set A obtained by selecting each element of $\{0,1\}^n$ independently with probability $\delta = 2\varepsilon$.

The probability that this set is smaller than $\varepsilon 2^n$ is at most (using the Chernoff bound [1]) $e^{-\frac{\varepsilon}{4}2^n}$.

Also, for any fixed d-dimensional projection P, the probability that P is contained in A is δ^{2^d}. The number of such projections is at most $(2d+2)^n$. Thus, the probability that such a randomly generated A has sufficiently large size and does not contain any d-dimensional projection is at least

$$1 - (2\varepsilon)^{2^d}(2d+2)^n - e^{-\frac{\varepsilon}{4}2^n}.$$

For sufficiently large n, this probability is greater than zero, and thus there exists such a set A with $|A| \geq 2^{-\frac{n\log(2d+2)}{2^d}-2}2^n$ which contains no d-dimensional projections. □

The bounds from the lemmas are summarized in the following theorem.

THEOREM 3.

$$2^{-\frac{n\log(2d+2)}{2^d}-2} \leq \rho(n,d) \leq 2^{-\frac{n}{d(2^d-1)}+1}.$$

COROLLARY 4. *Let d_n be the largest value such that every A with $|A| \geq 2^{n-1}$ contains a projection of dimension d_n.*

$$\log n - \log\log n + o(1) \leq d_n \leq \log n + \log\log\log n + o(1)$$

for all sufficiently large n.

For Boolean functions f, we state the following corollary.

COROLLARY 5. *If f is a Boolean function on n variables, then f has a monochromatic projection of dimension at least* $\log n - \log\log n + o(1)$.

Using a probabilistic argument very similar to the one used in Lemma 2.4, we obtain the following.

COROLLARY 6. *There are Boolean functions whose largest monochromatic projection has size at most* $\log n + \log\log\log n + o(1)$ *for all sufficiently large n.*

3. Explicit constructions. Although we showed that high density sets with no large projections exist, the question of constructing such sets remains open. In this section, we give some constructions of sets which contain no large projections. We first show that the set of codewords of a good code has the property that it contains no projections of greater than constant dimension. However, these sets have very low density, and it is not clear how to extend this technique to construct sets with high density but no large projections. Instead, we show that a randomly chosen low

degree GF(2) polynomial is not constant on any large dimensional projection, with high probability. While this does not give an explicit construction of a set with the desired properties, it does show that there exist easily computed sets which have no large projections.

3.1. Explicit constructions using codes. We start with a simple observation: if a set A contains a d-dimensional projection, then the set A has two points at a Hamming distance of at most n/d: if P is a d-dimensional projection, then it must contain a part with at most n/d variables, and by fixing all the variables outside the part consistent with the projection, we get two points which are at a distance of at most n/d. If A is a set of codewords for a code with rate r and distance δ, then A has size 2^{rn} and cannot contain a projection of dimension larger than n/δ. We can use constructions of linear codes to come up with "dense" sets with no large projections [8]. For example, for $0 < r < 1$, Justesen codes can be constructed with rate r and minimum distance at least $c_r n$ for some constant c_r for infinitely many n. Such codes can only include projections of bounded dimension.

For sets with constant density, it is easy to see that they can have at most constant minimum distance. Thus, this technique does not allow us to construct sets of size $c2^n$ which is guaranteed not to have projections of size $o(n)$.

3.2. Projections in functions defined by low degree GF(2) polynomials. The results of the previous section leave open the question of constructing sets of size $c2^n$ with no large projections. Moreover, it is not clear how to apply the ideas in the previous section to construct Boolean functions with no large monochromatic projections. In particular, it is an interesting open question to construct Boolean functions whose largest monochromatic projection has dimension $O(\log n)$. Although we fall short of answering this question, we show that there are simple objects which define Boolean functions without large monochromatic projections. In particular, we consider Boolean functions defined by GF(2) polynomials and estimate the dimension of the largest monochromatic projection as a function of degree of the polynomial. Let $\delta(d)$ denote the largest degree such that all polynomials of smaller degree have a monochromatic d-dimensional projection. We provide almost tight upper and lower bounds on $\delta(d)$. From these bounds, it follows that there are $\lceil \log n + \log\log\log n + o(1)\rceil$-degree GF(2) polynomials such that the corresponding Boolean functions have no monochromatic projections of dimension larger than $\log n + \log\log\log n$.

Let $f(x_1, \ldots, x_n)$ be a GF(2) polynomial of degree q. Let P be projection of dimension d, and let $\{y_1, \ldots, y_d\}$ be a set of representative free variables for P. To restrict a polynomial to a projection, replace each variable by the corresponding representative free variable or its negation, as appropriate. It is clear that the polynomial f, when restricted to the projection P, is a polynomial in $\{y_i\}$ of degree at most q. The following lemma shows that there exist low-degree polynomials which do not have large monochromatic projections. A special case of this lemma appears in [5].

LEMMA 3.1. *For $d \geq \lceil \log n + \log\log\log n + o(1)\rceil$ and all sufficiently large n,*

$$\delta(d) \geq Q_1(d),$$

where $Q_1(d)$ is the least integer such that $\sum_{i=0}^{Q_1(d)} \binom{d}{i} > n \log(2d+2) + 1$.

Proof. Let $q = Q_1(d)$. Also, fix a projection P of dimension d. Let $V_1 = \{x_1, \ldots, x_d\}$ be a set of representative free variables for P, and let V_2 be the set of all other variables.

Consider the following method of generating random elements from the space of GF(2) polynomials in the variables V_1 of degree at most q: select a polynomial

uniformly at random from the space of all GF(2) polynomials in variables $V_1 \cup V_2$ of degree at most q, then restrict it to the projection P. The polynomials over the variables V_1 correspond to the cosets of the additive group of polynomials which are zero when restricted to P. Therefore, it is easy to see that this distribution is uniform in the space of polynomials in variables V_1 of degree at most q. Hence, the probability that a randomly chosen polynomial is constant when restricted to the projection P is at most $2^{1-\sum_{i=0}^{q}\binom{d}{i}}$.

Since there are at most $(2d+2)^n$ projections of dimension d, the probability that a randomly chosen GF(2) polynomial of degree at most q has any monochromatic projection of dimension d is at most

$$(2d+2)^n 2^{1-\sum_{i=0}^{q}\binom{d}{i}}.$$

Given the definition of q, it follows that this probability is less than 1. Thus, there exists a polynomial of degree at most q which has no monochromatic projection of dimension d. □

COROLLARY 7. *There exists a degree q GF(2) polynomial whose largest monochromatic projection has dimension $O(q(n\log n)^{1/q})$.*

At the extreme end, we have the following.

COROLLARY 8. *There exists a GF(2) polynomial of degree $\lceil \log n + \log\log\log n + o(1)\rceil$ which has no monochromatic projection of dimension greater than $\lceil \log n + \log\log\log n + o(1)\rceil$.*

As far as dense sets with no large projections are concerned, we obtain the following corollary.

COROLLARY 9. *There exists a set of size 2^{n-1} defined by a degree d GF(2) polynomial which does not contain projections of dimension greater than $O(q(n\log n)^{1/q})$.*

We now show how to construct a monochromatic projection given an arbitrary low-degree polynomial.

THEOREM 4.

$$\delta(d) \le Q_2(d),$$

where $Q_2(d)$ is the greatest integer such that

$$2d+2+Q_2(d)\sum_{j=1}^{Q_2(d)-1}\binom{2d+3}{j} \le n.$$

Proof. Define $x^I = \prod_{x\in I} x_i$.

Let $f(x_1,\dots,x_n) = \sum_{I\subset[n]} a_I x^I$ be an arbitrary polynomial of degree at most $q = Q_2(d)$ which is not identically 1. We will construct a d-dimensional projection which is a subset of $\{(x_1,\dots,x_n)|f(x_1,\dots,x_n)=0\}$.

The projection will be constructed in several phases. Initially, all variables $V_0 = \{x_1,\dots,x_n\}$ are available and we have a projection P_0 where all variables are free. Let $R_0 = \emptyset$. During phase i, a nonempty set $A_i \subseteq V_{i-1}$ of available variables are equated among themselves to obtain a new projection P_i from P_{i-1}, and those variables become unavailable, that is, $V_i = V_{i-1} - A_i$. Then a representative free variable is selected from A_i and added to R_{i-1} to obtain R_i. At the end of each phase, we maintain the invariant that $f(x_1,\dots,x_n)$, when restricted to P_i, does not contain any monomials of degree 2 or more which involve only the variables from R_i.

We now select a nonempty set of available variables while maintaining the invariant. Assume we are at the beginning of phase $i+1$. By the induction hypothesis, the polynomial f restricted to P_i does not contain any monomials of degree 2 or higher involving only the variables in R_i. Let x_{r_i} be the representative variable for A_i, and define

$$f_i = \sum_{I \subseteq R_i} x^I \sum_{J \subseteq V_i, |I \cup J| \leq q} a_{I \cup J} x^J.$$

f_i is f restricted to P_i. We now select a nonempty subset of variables $A_{i+1} \subseteq V_i$ in the following way.

Let $I \subseteq R_i$ be such that $1 \leq |I| \leq q-1$ and

$$g_I = \sum_{J \subseteq V_i, 1 \leq |J| \leq q-|I|} a_{I \cup J} x^J.$$

g_I is a polynomial in the variables V_i and it is the coefficient of the term x^I in f_i except for the constant term. If $|I| \geq 2$, the constant coefficient of the term x^I in f_i is 0 by the induction hypothesis. If $|I| = 1$, then we will be dealing with a linear term which is not considered in the invariant. Note that if x is the characteristic vector of a set of variables to be chosen for A_{i+1} with representative variable $x_{r_{i+1}}$, then $g_I(x)$ is the coefficient of the term $x^I x_{r_{i+1}}$ when f is restricted to the projection P_{i+1}. Since g_I has no constant term, it evaluates to 0 when all the variables in V_i are set to 0. Now define

$$g = 1 + \prod_{I \subseteq R_i, 1 \leq |I| \leq q-1} (1 + g_I).$$

g is 0 exactly when all g_I are 0. The degree of g is at most $q \sum_{j=1}^{q-1} \binom{i}{j}$ and it is not identically equal to 1 since an assignment of 0's to the variables in V_i makes $g = 0$. If x is any nonzero solution of the equation $g = 0$, let A_{i+1} be the set of all variables in V_i which are set to 1 in x. Let P_{i+1} be the projection obtained from equating the variables in A_{i+1}. Also, update R_i to get R_{i+1} by adding a representative free variable for the class A_{i+1}. By the definition of g, f when restricted to P_{i+1} does not contain any monomials of degree 2 or more involving only the variables in R_{i+1}.

We now show that there is at least one nonzero solution to $g = 0$ with a "small" number of ones in the solution, thus ensuring that we can choose a small but nonempty set of variables to form the new part A_{i+1}. To show this, we use the following fact.

FACT 1. *Any GF(2) polynomial $T(x_1, \ldots, x_m)$ in m variables of degree at most $k < m$ which is not identically 1 must have a nonzero solution with at most $k+1$ ones.*

Proof. Find a maximal degree monomial M of T and select a variable which does not appear in M. Set this variable to 1 and set all other variables that do not appear in M to 0. After this assignment, T still contains the monomial M and thus is not identically 1. Hence, it has a solution containing at most k ones. Altogether, we have a nonzero solution with at most $k+1$ ones. □

Returning to the proof of the theorem, in step i, the degree of g is at most $q \sum_{j=1}^{q-1} \binom{i}{j}$ and so by Fact 1, there exists a solution with at most $1 + q \sum_{j=1}^{q-1} \binom{i}{j}$ ones. We continue this process, selecting A_i at each step, until there are no longer enough variables remaining. Let P_t be the final projection we obtain in this process. At this point, we make three modifications to P_t to obtain the desired projection. First,

all remaining available variables are set to 0. This ensures that f restricted to the projection has degree at most 1. Second, if f restricted to the projection has a nonzero constant term, set one of the free variables to 1. Finally, pair up all remaining free variables, and equate the variables of each pair (if the number of free variables is odd, set one free variable to 0) to get the final projection P. At this point, the polynomial f restricted to the projection P is identically 0. Moreover, P has at least $(t-2)/2$ free variables. We have

$$\sum_{i=1}^{2d+2}\left(1+q\sum_{j=1}^{q-1}\binom{i}{j}\right)=2d+2+q\sum_{j=1}^{q-1}\sum_{i=1}^{2d+2}\binom{i}{j}=2d+2+q\sum_{j=1}^{q-1}\binom{2d+3}{j}\le n$$

by the choice of q, thus guaranteeing $t \ge (2d+2)$. Therefore, P has at least d free variables, completing the proof of the theorem. □

COROLLARY 10. *Every degree q $GF(2)$ polynomial contains a monochromatic projection of dimension $\Omega(qn^{1/q})$.*

REFERENCES

[1] N. ALON, J. SPENCER, AND P. ERDŐS, *The Probabilistic Method*, Wiley-Interscience, New York, 1992.

[2] P. ERDŐS, *On extremal problems of graphs and generalized graphs*, Israel J. Math., 2 (1964), pp. 183–190.

[3] D.S. GUNDERSON, V. RÖDL, AND A. SIDORENKO, *Extremal problems for sets forming Boolean algebras and complete partite hypergraphs*, J. Combin Theory Ser. A, to appear.

[4] J. NEŠETŘIL, *Ramsey theory*, in Handbook of Combinatorics, Vols. 1, 2, R.L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 1331–1403.

[5] R. PATURI, M.E. SAKS, AND F. ZANE, *Exponential lower bounds for depth 3 Boolean circuits*, in Proc. Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997, ACM, New York, pp. 56–91.

[6] P. PUDLÁK, V. RÖDL, AND P. SAVICKÝ, *Graph complexity*, Acta Inform., 25 (1988), pp. 515–535.

[7] A.A. RAZBOROV, *Bounded depth formulae in the basis $\{\&, \oplus\}$ and some combinatorial problems*, in Complexity of Algorithms and Applied Mathematical Logic, S. I. Adian, ed., VINITI, Moscow, 1988, pp. 149–166 (in Russian).

[8] J.H. VAN LINT, *Introduction to Coding Theory*, 2nd ed., Springer-Verlag, New York, 1992.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 633–643, November 1998

008

k-NEIGHBORHOOD-COVERING AND -INDEPENDENCE PROBLEMS FOR CHORDAL GRAPHS*

SHIOW-FEN HWANG† AND GERARD J. CHANG‡

Abstract. Suppose $G = (V, E)$ is a simple graph and k is a fixed positive integer. A vertex z k-neighborhood-covers an edge (x, y) if $d(z, x) \leq k$ and $d(z, y) \leq k$. A k-neighborhood-covering set is a set C of vertices such that every edge in E is k-neighborhood-covered by some vertex in C. A k-neighborhood-independent set is a set of edges in which no two distinct edges can be k-neighborhood-covered by the same vertex in V. In this paper we first prove that the k-neighborhood-covering and the k-neighborhood-independence problems are NP-complete for chordal graphs. We then present a linear-time algorithm for finding a minimum k-neighborhood-covering set and a maximum k-neighborhood-independent set for a strongly chordal graph provided that a strong elimination ordering is given in advance.

Key words. k-neighborhood-covering, k-neighborhood-independence, chordal graph, strongly chordal graph, strong elimination order

AMS subject classifications. 05C80, 05C70, 68R10, 68Q20

PII. S0895480193244322

1. Introduction.

Domination is a natural model for location problems in operations research. This paper studies a variant of the domination problem we call k-neighborhood-covering. All graphs in this paper are simple, i.e., finite, undirected, loopless, and without multiple edges. In a graph $G = (V, E)$, the *length* of a path is the number of edges in the path. The *distance* $d(x, y)$ from vertex x to vertex y is the minimum length of a path from x to y; $d(x, y) = \infty$ if there is no path from x to y. A vertex x *k-dominates* another vertex y if $d(x, y) \leq k$. A vertex z *k-neighborhood-covers* an edge (x, y) if $d(z, x) \leq k$ and $d(z, y) \leq k$, or, equivalently, z k-dominates both x and y. A vertex set $C \subseteq V$ is a *k-neighborhood-covering set* if every edge in E is k-neighborhood-covered by some vertex in C. The *k-neighborhood-covering number* $\rho_N(G, k)$ of G is the minimum cardinality of a k-neighborhood-covering set. An edge set $I \subseteq E$ is a *k-neighborhood-independent set* if no two distinct edges in I are k-neighborhood-covered by the same vertex in V. The *k-neighborhood-independence number* $\alpha_N(G, k)$ of G is the maximum cardinality of a k-neighborhood-independent set.

For any graph G and any positive integer k, $\alpha_N(G, k)$ and $\rho_N(G, k)$ are related by the following obvious *max-min inequality*:

$$\alpha_N(G, k) \leq \rho_N(G, k).$$

For $k = 1$, Lehel and Tuza [11] proved that the above inequality is in fact an equality for odd-sun-free chordal graphs G and presented a linear-time algorithm for computing $\alpha_N(G, 1) = \rho_N(G, 1)$ for interval graphs. Wu [17] presented an $O(|V|^3)$-time algorithm for determining $\rho_N(G, 1) = \alpha_N(G, 1)$ for a strongly chordal graph G. Chang, Farber,

*Received by the editors February 16, 1993; accepted for publication (in revised form) September 6, 1997; published electronically September 1, 1998.
http://www.siam.org/journals/sidma/11-4/24432.html
†Department of Computer Science, Fen Chia University, Taichung 40724, Taiwan.
‡Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan (gjchang@math.nctu.edu.tw). The research of this author was supported in part by the National Science Council under grant NSC81-0208-M009-26 and in part by DIMACS.

and Tuza [6] presented linear-time algorithms for computing $\alpha_N(G,1) = \rho_N(G,1)$ for a strongly chordal graph G provided that a strong elimination order is known in advance. The purpose of this paper is to study $\alpha_N(G,k)$ and $\rho_N(G,k)$ for chordal graphs.

A graph is *chordal* (or *triangulated*) if every cycle of length greater than three has a chord, which is an edge joining two noncontiguous vertices in the cycle. In a graph $G = (V,E)$, the *neighborhood* $N(v)$ of a vertex v is the set of all vertices adjacent to v and the *closed neighborhood* $N[v] = N(v) \cup \{v\}$. A vertex v is *simplicial* if $N[v]$ is a clique. It is well known that a graph $G = (V,E)$ is chordal if and only if it has a *perfect elimination order*, i.e., an ordering $[v_1, v_2, \ldots, v_n]$ of V such that each v_i is a simplicial vertex of the subgraph G_i induced by $\{v_i, v_{i+1}, \ldots, v_n\}$ (see [8]).

An *s-sun* (or *incomplete trampoline of order s*) is a chordal graph having $2s$ vertices $a_1, a_2, \ldots, a_s, b_1, b_2, \ldots, b_s$ such that $(a_1, a_2, \ldots, a_s, a_1)$ is a cycle and each b_i has exactly two neighbors a_i and a_{i+1}, where $a_{n+1} = a_1$. A graph is *sun-free* (resp., *odd-sun-free*) if it contains no *s-sun* as a subgraph for all $s \geq 3$ (resp., for all odd $s \geq 3$) (see [4, 5]). Sun-free chordal graphs are called *strongly chordal graphs* in [7]. A vertex is *simple* if the set $\{N[u]\colon u \in N[v]\}$ can be linearly ordered by inclusion. Farber [7] proved that a graph $G = (V,E)$ is strongly chordal if and only if it has a *simple elimination order*, i.e., an ordering $[v_1, v_2, \ldots, v_n]$ of V such that each v_i is a simple vertex of G_i. A *strong elimination order* is a simple elimination order such that $N_{G_i}[v_j] \subseteq N_{G_i}[v_k]$ whenever $i \leq j \leq k$ and $v_j, v_k \in N_{G_i}[v_i]$. Anstee and Farber [2] presented an $O(|V|^3)$-time algorithm, Hoffman, Kolen, and Sakarovitch [9] presented an $O(|V|^3)$-time algorithm, Lubiw [12] presented an $O(L \log^2 L)$-time algorithm where $L = |V| + |E|$, Paige and Tarjan [13] presented an $O(L \log L)$-time algorithm, and Spinrad [16] presented an $O(|V|^2)$-time algorithm for finding a strong elimination order of a strongly chordal graph $G = (V,E)$.

The contents of this paper are as follows. In section 2, we prove that the k-neighborhood-covering and the k-neighborhood-independence problems are NP-complete for chordal graphs. Section 3 gives a linear-time algorithm for computing $\alpha_N(G,k)$ and $\rho_N(G,k)$ for a strongly chordal graph G provided that a strong elimination ordering is given in advance. Section 4 verifies the correctness of the algorithm. The algorithm in fact gives a k-neighborhood-covering set C^* and a k-neighborhood-independent set I^* with $|C^*| = |I^*|$. Consequently, C^* and I^* are optimal and $\alpha_N(G,k) = \rho_N(G,k)$ for any strongly chordal graph G.

In the rest of this section, we discuss the k-neighborhood-covering problem by means of the integer-linear programming method. The k-neighborhood-covering problem for a graph G is precisely the integer-linear programming problem

$$\text{(ILP)} \qquad \min \{1 \cdot x\colon Mx \geq 1 \text{ and } x \geq 0 \text{ integral}\},$$

where M is the 0-1 matrix whose rows are indexed by the edges of G and whose columns are indexed by the vertices and which possesses an entry of 1 for row e and column v if and only if v is within distance k of both ends of e. The k-neighborhood-independence problem is the following integer dual of (ILP):

$$\text{(ID)} \qquad \max \{y \cdot 1\colon yM \leq 1 \text{ and } y \geq 0 \text{ integral}\}.$$

We now claim that if G is strongly chordal, then matrix M is totally balanced. First, the 0-1 matrix A_k, with rows and columns both indexed by vertices of G and with a 1 for row u and column v if and only if u and v are within distance k, is totally balanced [12]. Second, the property of being totally balanced is preserved by the operation of

adding a new row which is the intersection of two existing rows [1]. Thus for each edge (u, v) we can add a row which is the intersection of row u and row v—this new row has 1's for precisely the vertices x that are within distance k of both u and v. Finally, deleting the rows corresponding to vertices yields the matrix M.

Since M is totally balanced, the integrality conditions in (ILP) and (ID) can be dropped, thus giving a min-max equality and a polynomial-time algorithm by means of a greedy method [9]. This method also works for the weighted cases of the problems. The drawback of the method is that it takes more than linear-time to compute M. The main effort of this paper is to give a linear-time algorithm to solve these problems without explicit computation of M. This is similar to the case of solving the k-domination problem in strongly chordal graphs in linear-time without taking the k-power of the graph; see [3].

2. NP-completeness for chordal graphs. In this section we show that for any fixed k, the k-neighborhood-covering and the k-neighborhood-independence problems are NP-complete for chordal graphs. We shall do this by reducing the problems with $k = 1$ for split graphs, which are known to be NP-complete in [6], to the problems for chordal graphs. A graph $G = (V, E)$ is *split* if its vertex set V is the disjoint union of a clique C and an independent set S. Split graphs are chordal.

THEOREM 2.1. *For any fixed positive integer k, the k-neighborhood-covering and the k-neighborhood-independence problems are NP-complete for chordal graphs.*

Proof. For any split graph $G = (V, E)$, whose vertex set V is the disjoint union of a clique C and an independent set S, construct the graph $G_k = (V_k, E_k)$ by attaching a path $s \equiv s_1, s_2, \ldots, s_k$ of length $k - 1$ to each vertex s in S. In other words,

$$V_k = C \cup \{s_i\colon\ s \in S \text{ and } 1 \le i \le k\} \quad \text{and} \quad E_k = E \cup \{(s_i, s_{i+1})\colon\ 1 \le i \le k-1\},$$

where s_1 is considered to be the same as s, G_k is clearly a chordal graph. We shall prove that $\rho_N(G_k, k) = \rho_N(G, 1)$ and $\alpha_N(G_k, k) = \alpha_N(G, 1)$. If these two equalities hold, then the theorem follows from the fact that the 1-neighborhood-covering and the 1-neighborhood-independence problems are NP-complete for split graphs (see [6]).

Suppose D is a minimum 1-neighborhood-covering set of G. Any edge in E is 1-neighborhood-covered and so k-neighborhood-covered by some vertex in D. For any edge $(s_i, s_{i+1}) \in E_k - E$, there exists some $x \in D$ such that $d(x, s) \le 1$. So $d(x, s_i) \le i < k$ and $d(x, s_{i+1}) \le i + 1 \le k$, i.e., x k-neighborhood-covers (s_i, s_{i+1}). Therefore D is a k-neighborhood-covering set of G_k. This gives $\rho_N(G_k, k) \le \rho_N(G, 1)$.

Conversely, suppose D is a minimum k-neighborhood-covering set of G_k. We can assume that $D \subseteq C$, otherwise any s_i in D can be replaced by a vertex adjacent to $s \equiv s_1$ in C to form a new minimum k-neighborhood-covering set. Consider any edge (x, y) in E. If x and y are both in C, then (x, y) is clearly 1-neighborhood-covered by any vertex in D. Suppose $x \in C$ and $y \in S$. Since D is a k-neighborhood-covering set of G_k, there exists some vertex z in D that k-neighborhood-covers (y_{k-1}, y_k). Both $z \in C$ and $d(z, y_k) \le k$ imply that z is adjacent to y and so z 1-neighborhood-covers (x, y). Therefore D is a 1-neighborhood-covering set of G. This gives $\rho_N(G_k, k) \ge \rho_N(G, 1)$. Together these inequalities imply that $\rho_N(G_k, k) = \rho_N(G, 1)$.

Suppose I is a minimum 1-neighborhood-independent set of G. It must be the case that each edge of I has one end in C and the other end in S. Also $S' = \{s \in S$: there is some $(x, s) \in I\}$ is a 2-independent set; i.e., no two distinct vertices of S' have a common neighbor. So $I' = \{(s_{k-1}, s_k)\colon\ s \in S'\}$ is a k-neighborhood-independent set of G_k. This gives $\alpha_N(G_k, k) \ge \alpha_N(G, 1)$.

Conversely, suppose I' is a minimum k-neighborhood-independent set G_k. Each edge of I' must be of the form (s_{k-1}, s_k) for some $s \in S$. Also $S' = \{s \in S$: some $(s_{k-1}, s_k) \in I'\}$ is a 2-independent set. For any $s \in S'$, choose a neighbor s' of s in C. Then $I = \{(s', s)$: $s \in S'\}$ is a 1-neighborhood-independent set of G. This gives $\alpha_N(G_k, k) \leq \alpha_N(G, 1)$. Together these inequalities imply that $\alpha_N(G_k, k) = \alpha_N(G, 1)$. □

3. The algorithm for strongly chordal graphs. In this section we set forth a linear-time algorithm for the k-neighborhood-covering and the k-neighborhood-independence problems for a strongly chordal graph G provided that a strong elimination ordering is given in advance. Without loss of generality, we may assume that G has no isolated vertices. The algorithm in fact gives a k-neighborhood-covering set C^* and a k-neighborhood-independent set I^* with $|C^*| = |I^*|$. By definitions and the max-min inequality,

$$|C^*| = |I^*| \leq \alpha_N(G, k) \leq \rho_N(G, k) \leq |C^*|.$$

Hence all inequalities are equalities. Consequently, C^* and I^* are optimal and $\alpha_N(G, k) = \rho_N(G, k)$ for any strongly chordal graph G.

Suppose $G = (V, E)$ is a strongly chordal graph for which a strong elimination order $[v_1, v_2, \ldots, v_n]$ is given. Note that this is also a perfect elimination order of the chordal graph G. For simplicity we identify vertex v_i as i, and hence $[1, 2, \ldots, n]$ is a strong elimination order. For any vertices i and j, $i \leq j$, let

$$N_i(j) = \{l \in: (j, l) \in E \text{ and } l \geq i\} \text{ and}$$
$$N_i[j] = N_i(j) \cup \{j\}.$$

As in [7], we use $N^+(i)$ for $N_i(i)$ and $N^+[i]$ for $N_i[i]$. A useful property of a chordal graph G, in which $[1, 2, \ldots, n]$ is a perfect elimination order, is that

$$N^+[i] \text{ is a clique for any } i. \tag{3.1}$$

A useful property of a strongly chordal graph G, in which $[1, 2, \ldots, n]$ is a strong elimination order, is that

$$j, l \in N^+[i] \text{ and } i \leq j \leq l \text{ imply } N_i[j] \subseteq N_i[l]. \tag{3.2}$$

(3.2) states that in the graph G_i induced by $\{i, i+1, \ldots, n\}$, the maximum neighbor of i has the largest closed neighborhood among all neighbors of i. So, the maximum neighbor is a most powerful dominating vertex. This is important to the development that follows. (3.1) and (3.2) are also used frequently in the proofs of lemmas and theorems in section 4.

The idea behind our algorithm for the k-neighborhood-covering and the k-neighborhood-independence problems is analogous to, but much more complicated than, that behind the algorithms for the k-domination problem (see [3, 15]). In fact, a $(k-1)$-dominating set is a k-neighborhood-covering set. However, the converse is not true. Note that a vertex set C is a k-neighborhood-covering set if and only if for any edge e in G, either one end vertex of e is $(k-1)$-dominated by some vertex in C, or both end vertices of e are exactly k-distant from the same vertex in C. So our algorithm retains the spirit of the $(k-1)$-domination problem with special attention to cases in which a *critical edge* e occurs; i.e., both end vertices of e are exactly k-distant from

the same vertex in C. To handle critical edges, we employ some of the ideas in [6] for 1-neighborhood-covering and -independence.

The algorithm processes vertices in the order $1, 2, \ldots, n$. Initially the k-neighborhood-covering set C and the k-neighborhood-independent set I are empty. In iteration i, the algorithm determines whether vertex i must be put into C. If the answer is positive, the algorithm also finds an edge to put into I. After processing, vertex i is deleted from the graph and $i+1$ becomes a simple vertex of the remaining graph.

For technical reasons, we associate each vertex i with two nonnegative integers $a(i)$ and $b(i)$, two vertices $A(i)$ and $s(i)$, and a subset $N_0^+(i)$ of $N^+(i)$. The meanings of these items are as follows. For each vertex i the algorithm must eventually include some vertex that is within distance $a(i)$ from i in the k-neighborhood-covering set C. At any time, there is a vertex in the current C that is within distance $b(i)$ from i. Both $a(i)$ and $b(i)$ keep decreasing as the algorithm proceeds. $a(i)$ decreases when i is the maximum neighbor of a smaller vertex i' that is not properly neighborhood-covered by a vertex of C within distance $a(i')$ in iteration i'. In this case, $a(i)$ is set to $a(i') - 1$. Similarly, in a previous iteration, $a(i')$ was set to $a(i'') - 1$. Continuing this argument, there exists a smallest vertex i^* that forces $\ldots, i''$, i', i to decrease their $a(\cdot)$ values, although $a(i^*)$ never changes. We use $A(i)$ to denote this *initial vertex* i^* from which $a(i)$ decreases. $s(i)$ is the optimal candidate for j such that (i, j) is put into I. $N_0^+(i)$ is a set of candidates for $s(i)$. More precisely, $N_0^+(i)$ contains those $j \in N^+(i)$ such that there is no vertex i' 1-neighborhood-covering (i, j) and i' itself is $(k-1)$-dominated by a vertex in the current C. $s(i)$ is chosen to be min $N_0^+(i)$ in iteration i. If in iteration i the algorithm determines that i should be put into C, it will also put the edge $(A(i), s(A(i)))$ into I. Initially, $a(i) = k$, $b(i) = \infty$, $A(i) = i$, $s(i) = \infty$, and $N_0^+(i) = N^+(i)$ for all $i \in V$. The algorithm processes vertex i according to one of the following cases.

When $a(i) = 0$, vertex i must be put into C. When $a(i) < b(i)$ and $N^+(i) = \emptyset$, all vertices in the current C are farther than distance $a(i)$ from i and no vertex $j > i$ is adjacent to i. So, we need to put i into C. In these two cases, we also put the edge $(A(i), s(A(i)))$ into I. Since i is now in C, $b(i)$ is set to 0.

Suppose $a(j) \geq a(i)$ and $b(j) \geq a(i)$ for all $j \in N^+(i)$. When $0 < a(i) < b(i)$ and $N^+(i) \neq \emptyset$, all vertices in the current C are farther than distance $a(i)$ from i. In this case, we need to find a vertex no farther than $a(i) - 1$ from the maximum neighbor m of i, since for any vertex $j > i$ there is a shortest path from j to i that passes through m. When $a(i) = b(i) = k$ and $N_0^+(i) \neq \emptyset$, there is one vertex in the current C that is k-distant from vertex i. However, this vertex does not k-neighborhood-cover any edge (i, j) with $j \in N_0^+(i)$. Again, we need to find a vertex that $(k-1)$-dominates m and so k-neighborhood-covers all edges (i, j) with $j \in N_0^+(i)$.

When $a(i) \geq b(i)$ and $k > b(i)$, there is a vertex in the current C that is within distance $a(i)$ from i. This vertex in fact $(k-1)$-dominates i and so k-neighborhood-covers all edges incident to i. When $a(i) = b(i) = k$ and $N_0^+(i) = \emptyset$, by the definition of $N_0^+(i)$, any edge (i, j) with $j \in N^+(i)$ is k-neighborhood-covered by some vertex in C. When $a(j) < a(i)$ or $b(j) < a(i)$ for some $j \in N^+(i)$, a vertex of C that is within distance $a(j)$ from j must be within distance $a(i)$ from i. In these three cases, we do not need to do anything.

Finally, we need to update $b(j)$ and $N_0^+(j)$ for all vertices $j \in N^+(i)$.

We can summarize the procedure described above in the following algorithm:

Algorithm CI.

Input. A strongly chordal graph $G = (V, E)$ without isolated vertices and in which

$[1, 2, \ldots, n]$ is a strong elimination order.
Output. A minimum k-neighborhood-covering set C and a maximum k-neighborhood-independent set I of G.
Method.

```
1.   C ⟵ ∅;  I ⟵ ∅;
2.   for all i ∈ V do
3.       [a(i) ⟵ k;  b(i) ⟵ ∞;  A(i) ⟵ i;  s(i) ⟵ ∞;  N0+(i) ⟵ N+(i)];
4.   for i = 1 to n do
5.       Case 1: a(i) = 0 or (a(i) < b(i) and N+(i) = ∅)
6.           C ⟵ C ∪ {i};
7.           I ⟵ I ∪ {(A(i), s(A(i)))};
8.           b(i) ⟵ 0;
9.       Case 2: ((0 < a(i) < b(i) and N+(i) ≠ ∅) or (a(i) = b(i) = k
                  N0+(i) ≠ ∅)) and (a(j) ≥ a(i) and b(j) ≥ a(i) for all j ∈ N+(i))
11.          m ⟵ max N+(i);
12.          a(m) ⟵ a(i) − 1;
13.          A(m) ⟵ A(i);
14.          s(i) ⟵ min N0+(i);      {where min ∅ = ∞}
15.      Case 3: (a(i) ≥ b(i) and k > b(i)) or (a(i) = b(i) = k and N0+(i) = ∅)
16.                  or (a(j) < a(i) or b(j) < a(i) for all j ∈ N+(i))
17.          do nothing;
18.      for all j ∈ N+(i) do b(j) ⟵ min{b(j), b(i) + 1};
19.      R ⟵ {j ∈ N0+(i) : a(j) = b(j) = b(i) + 1 = k};
20.      for all j ∈ R do N0+(j) ⟵ N0+(j) − R;
21.      end for.
```

4. Correctness of the algorithm. This section proves the correctness of Algorithm CI. First, we note that during the execution of the algorithm, $a(i)$, $b(i)$, $A(i)$, $s(i)$, C, and I are updated. We shall denote their final values by $a^*(i)$, $b^*(i)$, $A^*(i)$, $s^*(i)$, C^*, and I^*, respectively. Note that $a(i)$, $b(i)$, and $A(i)$ keep decreasing and stay at their final values from the beginning of iteration i. $s(i)$ only changes from ∞ to $\min N_0^+(i)$ in iteration i when Case 2 of the algorithm holds.

Our proof of the correctness of Algorithm CI is based on proving that C^* is a k-neighborhood-covering set (Theorem 4.4), I^* is a k-neighborhood-independent set (Theorem 4.11), and $|C^*| = |I^*|$ (Theorem 4.7). If these conditions hold, C^* is a minimum k-neighborhood-covering set, I^* is a maximum k-neighborhood-independent set, and $\alpha_N(G, k) = \rho_N(G, k)$.

LEMMA 4.1. *For any vertex $x \in V$, there exists some $y \in C^*$ such that $d(x, y) \le b^*(x)$.*

Proof. We claim that in any iteration, for any vertex $x \in V$, there exists some $y \in C$ such that $d(x, y) \le b(x)$. Initially, $b(x) = \infty$ and $C = \emptyset$. The claim holds if we interpret it to be $\min_{y \in C} d(x, y) \le b(x)$ and $\min \emptyset = \infty$. $b(x)$ only changes its value in lines 8 and 18. If $b(x)$ is reset to 0 in line 8, then x is added to C in line 6. So $d(x, x) = 0 = b(x)$ for $x \in C$. Suppose $b(x)$ is reset to $b(i) + 1$ in line 18 for $x \in N^+(i)$ and $b(x) > b(i) + 1$; it then follows from the induction hypothesis that there must be some $y \in C$ such that $d(i, y) \le b(i)$. Therefore, $d(x, y) \le d(x, i) + d(i, y) \le 1 + b(i) < b(x)$. This proves the claim. Consequently, the lemma holds. □

LEMMA 4.2. *For any vertex $x \in V$, one of the following three statements holds.*

(1) $b^*(x) \le a^*(x)$.
(2) $b^*(j) < a^*(x)$ *for some* $j \in N^+(x)$.

(3) $a^*(j) < a^*(x)$ *for some* $j \in N^+(x)$.

Proof. We shall prove the lemma by considering the following cases in iteration x.

If Case 1 of the algorithm holds, then $b^*(x) = 0 \leq a^*(x)$; i.e., (1) holds.

If Case 2 of the algorithm holds, then there exists $j = \max N^+(x)$ such that $a(j) = a(x) - 1$ and so $a^*(j) \leq a(j) < a(x) = a^*(x)$; i.e., (3) holds.

If Case 3 of the algorithm holds, then $b(x) \leq a(x)$ or there exists $j \in N^+(x)$ such that $a(j) < a(x)$ or $b(j) < a(x)$. For the first case, $b^*(x) \leq b(x) \leq a(x) = a^*(x)$; i.e., (1) holds. For the second case, $a^*(j) \leq a(j) < a(x) = a^*(x)$; i.e., (3) holds. For the third case, $b^*(j) \leq b(j) < a(x) = a^*(x)$; i.e., (2) holds. □

LEMMA 4.3. *For any vertex* $x \in V$, *there exists some* $y \in C^*$ *such that* $d(x, y) \leq a^*(x)$.

Proof. Repeatedly apply Lemma 4.2 to get a sequence $x \equiv x_0, x_1, \ldots, x_{r-1}, x_r$ such that $x_i \in N^+(x_{i-1})$ for $1 \leq i \leq r$ and

$$b^*(x_r) \leq a^*(x_r) < a^*(x_{r-1}) < \cdots < a^*(x_1) < a^*(x_0) = a^*(x)$$

or
$$b^*(x_r) < a^*(x_{r-1}) < \cdots < a^*(x_1) < a^*(x_0) = a^*(x).$$

Then $r + b^*(x_r) \leq a^*(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(x_r, y) \leq b^*(x_r)$. Hence $d(x, y) \leq d(x, x_r) + d(x_r, y) \leq r + b^*(x_r) \leq a^*(x)$. □

THEOREM 4.4. C^* *is a* k*-neighborhood-covering set of* G.

Proof. Suppose (x, z) is an edge with $x < z$, i.e., $z \in N^+(x)$. We shall prove the theorem according to (1), (2), and (3) of Lemma 4.2.

(1) $b^*(x) \leq a^*(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(x, y) \leq b^*(x) \leq a^*(x) \leq k$. If $d(x, y) \leq k - 1$, then y k-neighborhood-covers (x, z). So we may assume that $d(x, y) = b^*(x) = a^*(x) = k$. Since Lemma 4.2 (1) holds only when Case 2 of the algorithm does not, $N_0^+(x) = \emptyset$ in iteration x. By $z \in N^+(x)$ and the updating rule for $N_0^+(j)$ in lines 18 and 19, there exists an i such that $x, z \in N_0^+(i)$ and $b(i) = k - 1$ in iteration i. By Lemma 4.1, there exists some $y \in C^*$ such that $d(i, y) \leq b(i) = k-1$. Hence y $(k-1)$-dominates i and i 1-neighborhood-covers (x, z); i.e., y k-neighborhood-covers (x, z).

(2) $b^*(j) < a^*(x)$ for some $j \in N^+(x)$. By Lemma 4.1, there exists some $y \in C^*$ such that $d(j, y) \leq b^*(y) < a^*(x) \leq k$; i.e., y $(k-1)$-dominates j. Since $z, j \in N^+(x)$, j 1-neighborhood-covers (x, z) by (3.1). So y k-neighborhood-covers (x, z).

(3) $a^*(j) < a^*(x)$ for some $j \in N^+(x)$. By Lemma 4.3, there exists some $y \in C^*$ such that $d(j, y) \leq a^*(y) < a^*(x) \leq k$; i.e., y $(k-1)$-dominates j. Since $z, j \in N^+(x)$, j 1-neighborhood-covers (x, z) by (3.1). So y k-neighborhood-covers (x, z). □

LEMMA 4.5. *If* $y \in C^*$, *then* $a^*(y) < k$.

Proof. Since $y \in C^*$, Case 1 of the algorithm holds in iteration y, i.e., $a(y) = 0$ or $a(y) < b(y)$ with $N^+(y) = \emptyset$. For the former case, clearly, $a^*(y) < k$. For the latter case, since G has no isolated vertex, there exists a largest neighbor w of y. Note that $y \in N^+(w)$. Suppose $N^+(w)$ contains a vertex y' other than y. Then by (3.1) y' is adjacent to y. So y' is a neighbor of y that is larger than w, which is a contradiction. Thus $N^+(w) = \{y\}$ and $y = \max N^+(w)$. We now consider the following cases in iteration w.

If Case 1 of the algorithm holds, then by line 8 $b(w) = 0$ and so, by line 18, $b(y) \leq 1$. Thus $b(y) \leq 1$ in iteration y and so $a^*(y) \leq a(y) < b(y) \leq 1 \leq k$.

If Case 2 of the algorithm holds, then $a^*(y) \leq a(y) = a(w) - 1 < k$.

If Case 3 of the algorithm holds, then $b(w) < k$ or $a(w) = b(w) = k$ with $N_0^+(w) = \emptyset$ or $a(y) < a(w)$ or $b(y) < a(w)$. For the first case, $b(y) \leq k$ by line 18.

$b(y) \le k$ still holds in iteration y and so $a^*(y) \le a(y) < b(y) \le k$. For the second case, by the updating rule in lines 19 and 20, there exists some i such that $w, y \in N_0^+(i)$ and $a(y) = b(y) = k$ in iteration i. Thus $b(y) \le k$ still holds in iteration y and so $a^*(y) \le a(y) < b(y) \le k$. For the third case, $a^*(y) \le a(y) < a(w) \le k$. For the fourth case, $b(y) < a(w)$ still holds in iteration y and so $a^*(y) \le a(y) < b(y) < a(w) \le k$. □

LEMMA 4.6. *For any $\bar{x} \in C^*$ with $A(\bar{x}) = x$, there exists a unique increasing path $x \equiv \bar{x}_0, \bar{x}_1, \ldots, \bar{x}_u \equiv \bar{x}$ from x to $\bar{x}$ (with $u \ge 1$) such that $\bar{x}_i = \max N^+(\bar{x}_{i-1})$ for $1 \le i \le u$ and $A^*(\bar{x}_i) = x$ and $a^*(\bar{x}_i) = k - i$ for $0 \le i \le u$. Furthermore, $s^*(x) \ne \infty$ and $A^*(y) \ne A^*(\bar{x})$ for any $y \in C^* - \{\bar{x}\}$. Also, $u = k$ when $N^+(\bar{x}) \ne \emptyset$.*

Proof. Let $u = k - a^*(\bar{x})$ and $\bar{x}_u = \bar{x}$, i.e., $a^*(\bar{x}_u) = k - u$. By Lemma 4.5, $u \ge 1$. Note that $a^*(\bar{x}_u)$ is initially k and decreases only when Case 2 of the algorithm holds in some iteration i and $\bar{x}_u = \max N^+(i)$. There may be several such i. Let $\bar{x}_{u-1}$ be the maximum of all such i. In this case, $\bar{x}_u = \max N^+(\bar{x}_{u-1})$, $A^*(\bar{x}_u) = A^*(\bar{x}_{u-1})$, and $a^*(\bar{x}_{u-1}) = a^*(\bar{x}_u) + 1 = k - (u-1)$. Continuing the same argument, we get an increasing path $x \equiv \bar{x}_0, \bar{x}_1, \ldots, \bar{x}_u \equiv \bar{x}$ such that $\bar{x}_i = \max N^+(\bar{x}_{i-1})$ for $1 \le i \le u$, $A^*(\bar{x}_0) = A^*(\bar{x}_1) = \ldots = A^*(\bar{x}_u)$, and $a^*(\bar{x}_i) = k - i$ for $0 \le i \le k$. Since $a^*(\bar{x}_0) = k$, $A^*(\bar{x}_0)$ keeps its original value x, i.e., $A^*(\bar{x}_i) = x$ for all $0 \le i \le u$.

Furthermore, since $\bar{x}_i$ is uniquely determined by $\bar{x}_{i-1}$, such a path is unique and there is no $y \in C^* - \{\bar{x}\}$ with $A^*(y) = A^*(\bar{x})$.

Since $a^*(x) = k$, $a(x) = k$ at any time. In iteration x, Case 2 holds, i.e., $0 < a(x) < b(x)$ with $N^+(x) \ne \emptyset$ or $a(x) = b(x) = k$ with $N_0^+(x) \ne \emptyset$. For the latter case, $s^*(x) \ne \infty$ by line 14. For the former case, $b(x) > k$ in iteration x and so $b(x) > k$ in any previous iteration. By the definition of R in line 18, $x \notin R$ in any previous iteration. Hence by line 19, $N_0^+(x) = N^+(x) \ne \emptyset$ in iteration x. Again, $s^*(x) \ne \infty$.

If $N^+(\bar{x}) \ne \emptyset$, then $a(\bar{x}) = 0$ in line 5 in iteration $\bar{x}$. So, $k - u = a^*(\bar{x}_u) = a^*(\bar{x}) = 0$, i.e., $u = k$. □

We shall call the unique path $x \equiv \bar{x}_0, \bar{x}_1, \ldots, \bar{x}_u \equiv \bar{x}$ from x to $\bar{x}$ (with $u \ge 1$) in Lemma 4.6 the *maximum path* for $\bar{x}$. For any $\bar{x} \in C^*$, the corresponding edge $(x, x') \in I^*$, where $x = A^*(\bar{x})$ and $x' = s^*(x)$. Note that Case 2 holds in iteration $\bar{x}_i$ for $0 \le i \le u - 1$ and Case 1 holds in iteration $\bar{x}_u \equiv \bar{x}$. Also, in Lemma 4.6, instead of saying $a^*(\bar{x}_i) = k - i$, we can say $a(\bar{x}_i) = k - i$ in iteration $\bar{x}_{i-1}$ when line 12 is executed for $1 \le i \le u$. $\bar{x}_1, x' \in N^+(x)$ imply that $d(\bar{x}_1, x') \le 1$ by (3.1) and so $\bar{x}_1$ 1-neighborhood-covers (x, x'). This together with $d(\bar{x}_1, \bar{x}_p) \le p - 1$ implies that $\bar{x}_p$ p-neighborhood-covers (x, x') for $1 \le p \le u$.

THEOREM 4.7. $|C^*| = |I^*|$.

Proof. By Lemma 4.6, for any two distinct vertices y, $y' \in C^*$, $A^*(y) \ne A^*(y')$, $s(A^*(y)) \ne \infty$, and $s(A^*(y')) \ne \infty$. Hence the theorem holds, because each time a new vertex is added to C in line 6 a new edge is added to I in line 7. □

LEMMA 4.8. *If vertex z k-neighborhood-covers an edge (x, x'), then there exists a shortest x-z path $x \equiv x_0, x_1, \ldots, x_r \equiv z$ such that $r \le k$ and each x_p $(k-1)$-neighborhood-covers (x, x') for $1 \le p \le r - 1$.*

Proof. The lemma is easy when $d(x, z) = 1 + d(x', z)$ or $d(x', z) = 1 + d(x, z)$. For the case in which $d(x, z) = d(x', z) \le k$, the lemma follows from the fact that there exists a vertex w adjacent to both x and x' such that $d(w, z) = d(x, z) - 1$. (See [10, Lemma 1 (d)].) □

LEMMA 4.9. *Any shortest x_0-x_r path $x_0, x_1, \ldots, x_r$ is unimodal, i.e., $x_0 < x_1 < \cdots < x_{i-1} < x_i$ and $x_i > x_{i+1} > \cdots > x_r$ for some $0 \le i \le r$.*

Proof. If the sequence is not unimodal, then $x_{i-1}, x_{i+1} \in N^+(x_i)$ for some i with $1 \le i \le r - 1$. By Property (3.1), $x_{i-1}x_{i-1} \in E$ and so $x_0, x_1, \ldots, x_r$ is not a shortest

path. □

LEMMA 4.10. *If $y_0, y_1, \ldots, y_r$ is an increasing path and $y_0 \in C^*$, then $b(y_i) \leq i$ after iteration y_{i-1} for $1 \leq i \leq r$. Consequently, $b(y_i) \leq i$ in and after iteration y_i for $0 \leq i \leq r$.*

Proof. We shall prove the lemma by induction on i. If $i = 1$, then in iteration y_0, $b(y_0) = 0$ by line 8 of the algorithm and so $b(y_1) \leq 1$ by line 18. Suppose $i \geq 2$ and $b(y_{i-1}) \leq i - 1$ after iteration y_{i-2}. In iteration y_{i-1}, by line 18, we have $b(y_i) \leq b(y_{i-1}) + 1 \leq i$. Hence the lemma holds. □

THEOREM 4.11. *I^* is a k-neighborhood-independent set of G.*

Proof. Suppose I^* is not k-neighborhood-independent; i.e., there exists some vertex $z \in V$ that k-neighborhood-covers two distinct edges (x, x') and (y, y') in I^*, where $x' = s^*(x)$ and $y' = s^*(y)$. Without loss of generality, we may assume that z is set as large as possible.

By the algorithm, there exist $\bar{x}, \bar{y} \in C^*$ such that $x = A^*(\bar{x})$ and $y = A^*(\bar{y})$. Let $x \equiv \bar{x}_0, \bar{x}_1, \ldots, \bar{x}_u \equiv \bar{x}$ be the maximum path for $\bar{x}$ and $y \equiv \bar{y}_0, \bar{y}_1, \ldots, \bar{y}_v \equiv \bar{y}$ the maximum path for $\bar{y}$. Since $x \neq y$, $\bar{x} \neq \bar{y}$ by Lemma 4.6. Also, each $\bar{x}_p$ p-neighborhood-covers (x, x') for $1 \leq p \leq u$ and each $\bar{y}_q$ q-neighborhood-covers (y, y') for $1 \leq q \leq v$.

By Lemma 4.8, there exists a shortest x-z path P: $x \equiv x_0, x_1, \ldots, x_r \equiv z$ with $r \leq k$ and a shortest y-z path $y \equiv y_0, y_1, \ldots, y_t \equiv z$ with $t \leq k$ such that each x_p $(k-1)$-neighborhood-covers (x, x') for $1 \leq p \leq r-1$ and each y_q $(k-1)$-neighborhood-covers (y, y') for $1 \leq q \leq t-1$. By Lemma 4.9, $x_0 < x_1 < \cdots < x_i$ and $x_i > x_{i+1} > \cdots > x_r$ for some $0 \leq i \leq r$ and $y_0 < y_1 < \cdots < y_j$ and $y_j > y_{j+1} > \cdots > y_t$ for some $0 \leq j \leq t$.

Let i^* be the largest index such that $x_p = \bar{x}_p$ for $0 \leq p \leq i^*$. We may assume that i^* is as large as possible. Note that $i^* \leq i$. For the case of $i^* + 1 \leq i$, $i^* + 1 \leq u$, otherwise $u < i^* + 1$ would imply that $u < i \leq r \leq k$, but $x^*_{i+1} \in N^+(\bar{x}_u) = N^+(\bar{x})$ contradicts the last statement of Lemma 4.6. Since $x_{i^*+1}\bar{x}_{i^*+1} \in N^+(x_{i^*})$, by (3.1) $x_{i^*+1}\bar{x}_{i^*+1} \in E$. By the fact that $\bar{x}_{i^*+1} = \max N^+(\bar{x}_{i^*})$, we have $x_{i^*+1} < \bar{x}_{i_*+1}$. Now $i^* + 1 = i$, otherwise $i^* + 2 \leq i$ would imply that $\bar{x}_{i^*+1}, x_{i^*+2} \in N^+(x_{i^*+1})$, which in turn implies $\bar{x}_{i^*+1}x_{i^*+2} \in E$ and so P': $\bar{x}_0, \ldots, \bar{x}_{i^*}$, x_{i^*+1}, x_{i^*+2}, $x_{i^*+3} \ldots, x_r$ is a path with $i^*(P') > i^*(P)$. In conclusion, either $x_p = \bar{x}_p$ for $0 \leq p \leq i$ (when $i = i^*$) or $x_p = \bar{x}_p$ for $0 \leq p \leq i-1$ with $\bar{x}_{i-1} < x_i < \bar{x}_i$ form a clique of 3 vertices (when $i = i^* + 1$). Similarly, we may assume that either $y_q = \bar{y}_q$ for $0 \leq q \leq j$ (when $j = j^*$) or $y_q = \bar{y}_q$ for $0 \leq q \leq j-1$ with $\bar{y}_{j-1} < y_j < \bar{y}_j$ form a clique of 3 vertices (when $j = j^* + 1$).

We now claim that $z \neq \bar{y}$. Suppose, to the contrary, $z = \bar{y} \in C^*$. Suppose $i^* = i = r$. Since $\bar{x}_u = \bar{x} \neq \bar{y} = z = x_r = \bar{x}_{r^*}$, $r^* \leq u - 1$. But $\bar{x}_{r^*} \in C$ implies that Case 1 holds in iteration $\bar{x}_{i^*}$, in contradiction to the fact that Case 2 holds in iteration $\bar{x}_p$ for $0 \leq p \leq u - 1$. Therefore, either $i^* + 1 \leq i = r$ or $i < r$. In any case, $r > 0$. Suppose $i = 0$. Consider the increasing path $z \equiv x_r, x_{r-1}, \ldots, x_1, x_0 \equiv x \equiv \bar{x}_0$. By Lemma 4.10, $b(x_1) \leq r - 1$ in iteration x_1 and so, by line 18, $b(x_0) \leq b(x_1) + 1 \leq r$ in iteration x_1. Then $b(x) \leq r \leq k$ in iteration x. On the other hand, by Lemma 4.6, $a^*(x) = a(\bar{x}_0) = k$ and so, $a(x) = k$ in any iteration. Then $a(x) = b(x) = k$ in iteration x, since Case 2 of the algorithm holds. Thus, $b(x_1) = k - 1$ and $b(x) = k$ in iteration x_1, and $a(x) = b(x) = k = r$ in iteration x. By line 19, $x \in R$ in iteration x_1. Now, $d(z, x) = r = k$. Since $x' > x$, we also have $d(z, x') = k$. By Lemma 4.8, we may assume that x' is adjacent to x_1. By line 18, $b(x') \leq b(x_1) + 1 \leq k$ in iteration x_1

and hence $b(x') \le k$ in iteration x. Since Case 2 holds in iteration x, $a(x') \ge a(x) = k$ and $b(x') \ge a(x) = k$. So, $a(x') = b(x') = k$ in iterations x and x_1. By line 19, $x' \in R$ in iteration x_1. Therefore, by line 20, $x' \notin N_0^+(x)$ after iteration x_1 is completed. In iteration x, it is impossible that $x' = s(x) \equiv N_0^+(x)$. Thus $i \ge 1$.

First, consider the case of $i^* + 1 \le i = r$, i.e., $x_i = z = \bar{y} = \bar{y}_v \in C^*$. If $\bar{x}_{i-1} \le \bar{y}_{v-1}$, then $\bar{y}_{v-1} \in N_{\bar{x}_{i-1}}[x_i]$. By (3.2), $N_{\bar{x}_{i-1}}[x_i] \subseteq N_{\bar{x}_{i-1}}[\bar{x}_i]$. Thus, $\bar{y}_{v-1}$ is adjacent to $\bar{x}_i > x_i = \bar{y}_v$, in contradiction to $\bar{y}_v = \max N^+[\bar{y}_{v-1}]$. So, $\bar{y}_{v-1} < \bar{x}_{i-1}$. In iteration $\bar{y}_{v-1}$, $a(\bar{y}_v) = k - v$ by line 12 of the algorithm and Lemma 4.6. Also, $N^+(\bar{y}_v) \ne \emptyset$ since it contains $\bar{x}_i$. By Lemma 4.6, $v = k$. Then, $a(\bar{y}_v) = 0$ in and after iteration $\bar{y}_{v-1}$. In particular, $a(\bar{y}_v) = 0$ in iteration $\bar{x}_{i-1}$. But, in iteration $\bar{x}_{i-1}$, Case 2 holds. By line 9, $0 < a(\bar{x}_{i-1})$, but, by line 10, $0 = a(\bar{y}_v) \ge a(\bar{x}_{i-1})$, which is a contradiction.

Next, consider the case of $i < r$. If $\bar{x}_{i-1} \le x_{i+1}$, then x_{i+1} is adjacent to $\bar{x}_i$ by (3.2). Consider the increasing path $z \equiv x_r, x_{r-1}, \ldots, x_{i+1}, \bar{x}_i$. By Lemma 4.10, $b(\bar{x}_i) \le r - i \le k - i = a^*(\bar{x}_i)$ in iteration $\bar{x}_i$. When Case 1 holds in iteration $\bar{x}_i$, $a^*(\bar{x}_i) = 0$ and so $k = r = i$, which contradicts $i < r$. When Case 2 holds in iteration $\bar{x}_i$, $a(\bar{x}_i) = b(\bar{x}_i) = k$ and so $i = 0$, which contradicts $i \ge 1$. If $\bar{x}_{i-1} > x_{i+1}$, then $b(x_i) \le k - i$ after iteration x_{i+1} by Lemma 4.10. In particular, $b(x_i) \le k - i$ in iteration $\bar{x}_{i-1}$. But Case 2 of the algorithm holds in iteration $\bar{x}_{i-1}$. By line 10, $k - (i-1) = a(\bar{x}_{i-1}) \le b(x_i) \le k - i$, which is a contradiction.

So, in any case $z \ne \bar{y}$, i.e., $z < \bar{y}$. Similarly, $z < \bar{x}$. Consider the two increasing paths P_1 and P_2, where P_1 is $z \equiv x_r, x_{r-1}, \ldots, x_i, \bar{x}_{i^*}, \bar{x}_{i^*+1}, \ldots, \bar{x}_u \equiv \bar{x}$ (with $\bar{x}_{i^*}$ omitted when $x_i = \bar{x}_{i^*}$ and $i = i^*$) and P_2 is $z \equiv y_t, y_{t-1}, \ldots, y_j, \bar{y}_{j^*}, \bar{y}_{j^*+1}, \ldots, \bar{y}_v \equiv \bar{y}$ (with $\bar{y}_{j^*}$ omitted when $y_j = \bar{y}_{j^*}$ and $j = j^*$). By (3.1), $(\alpha, \beta) \in E$, where α is the second vertex of P_1 and β the second vertex of P_2. Note that $\alpha = x_p$ with $p \le r - 1$ or $\alpha = \bar{x}_p$ with $p \le u$, and $\beta = y_q$ with $q \le t - 1$ or $\beta = \bar{y}_q$ with $q \le v$. By Lemma 4.8 and the sentence just before Theorem 4.7, α $(k-1)$-neighborhood-covers (x, x') or β $(k-1)$-neighborhood-covers (y, y'), and so α or β k-neighborhood-covers both (x, x') and (y, y), contradicting the choice of z, except when $u = v = k$, $\alpha = \bar{x}_u$, $\beta = \bar{y}_v$, and $|P_1| = |P_2| = 1$. For the exceptional case, since either $i = i^*$ with $x_i = \bar{x}_i$ or $i = i^* + 1$ with $\bar{x}_{i-1} < x_i < \bar{x}_i$ form a clique, we have $r = i = i^* = u - 1 = k - 1$ or $r = i = i^* + 1 = u = k$ and so $\bar{x}_{k-1} \le z < \bar{x}_k$, i.e., $z, \bar{x}_k \in N^+[\bar{x}_{k-1}]$. Similarly, $z, \bar{y}_k \in N^+[\bar{y}_{k-1}]$. Without loss of generality, assume $\bar{x}_{k-1} \le \bar{y}_{k-1}$. Then, by (3.2), $\bar{y}_{k-1}$ is adjacent to $\bar{x}_k$. Now, $\bar{x}_k$ k-neighborhood-covers (x, x'). Also, $\bar{y}_{k-1}$ $(k-1)$-neighborhood-covers (y, y') and so $\bar{x}_k$ k-neighborhood-covers (y, y'). So we have $\bar{x}_k > z$ and $\bar{x}_k$ k-neighborhood-covers both (x, x') and (y, y'), which is a contradiction to the choice of z. □

THEOREM 4.12. *Algorithm CI finds a minimum k-neighborhood-covering set C^* and a maximum k-neighborhood-independent set I^* of a strongly chordal graph $G = (V, E)$ in linear-time if a strong elimination order is given.*

Proof. The correctness of the algorithm follows from Theorems 4.4, 4.7, and 4.11. The algorithm is linear, since iteration i costs only $O(|N^+(i)|)$ time excepting line 20, and line 20 in the whole algorithm costs at most $O(|E|)$ time. □

Acknowledgments. The authors thank referees for many constructive suggestions on the revision of this paper. In particular, one referee suggested the linear programming viewpoint as added at the end of section 1. A significant simplification of the proofs in section 4 was made after their comments.

REFERENCES

[1] R. P. Anstee, *Properties of* (0,1)*-matrices without certain configurations*, J. Combin. Theory Ser. A, 31 (1981), pp. 256–269.

[2] R. P. Anstee and M. Farber, *Characterization of totally balanced matrices*, J. Algorithms, 5 (1984), pp. 215–230.

[3] G. J. Chang, *Labeling algorithms for the domination problems in sun-free chordal graphs*, Discrete Appl. Math., 22 (1988/89), pp. 21–34.

[4] G. J. Chang and G. L. Nemhauser, *The k-domination and k-stability problems on sun-free chordal graphs*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 332–345.

[5] G. J. Chang and G. L. Nemhauser, *Covering, packing and generalized perfection*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 109–132.

[6] G. J. Chang, M. Farber, and Z. Tuza, *Algorithmic aspects of neighborhood numbers*, SIAM J. Discrete Math., 6 (1993), pp. 24–29.

[7] M. Farber, *Characterizations of strongly chordal graphs*, Discrete Math., 43 (1983), pp. 173–189.

[8] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[9] A. J. Hoffman, A. W. J. Kolen, and M. Sakarovitch, *Totally balanced and greedy matrices*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 721–730.

[10] R. Laskar and D. Shier, *Construction of* (r, d)*-invariant chordal graphs*, Congressus Numerantium, 33 (1981), pp. 155–165.

[11] J. Lehel and Z. Tuza, *Neighborhood perfect graphs*, Discrete Math., 61 (1986), pp. 93–101.

[12] A. Lubiw, *Doubly lexical ordering of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.

[13] R. Paige and R. E. Tarjan, *Three partition refinement algorithms*, SIAM J. Comput., 16 (1987), pp. 973–989.

[14] E. Sampathkumar and P. S. Neeralagi, *The neighborhood number of a graph*, Indian J. Pure Appl. Math., 16 (1985), pp. 126–132.

[15] P. J. Slater, *R-domination in graphs*, J. Assoc. Comput. Mach., 23 (1976), pp. 446–450.

[16] J. P. Spinrad, *Doubly lexical ordering of dense* 0-1 *matrices*, Inform. Process. Lett., 45 (1993), pp. 229–235.

[17] J. Wu, *Neighborhood-Covering and Neighborhood-Independence in Strongly Chordal Graphs*, manuscript.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 644–654, November 1998

009

LOCAL STRUCTURE WHEN ALL MAXIMAL INDEPENDENT SETS HAVE EQUAL WEIGHT*

YAIR CARO†, M. N. ELLINGHAM‡, AND J. E. RAMEY§

Abstract. In many combinatorial situations there is a notion of independence of a set of points. Maximal independent sets can be easily constructed by a greedy algorithm, and it is of interest to determine, for example, if they all have the same size or the same parity. Both of these questions may be formulated by weighting the points with elements of an abelian group, and asking whether all maximal independent sets have equal weight. If a set is independent precisely when its elements are pairwise independent, a graph can be used as a model. The question then becomes whether a graph, with its vertices weighted by elements of an abelian group, is *well-covered*, i.e., has all maximal independent sets of vertices with equal weight. This problem is known to be co-NP-complete in general. We show that whether a graph is well-covered or not depends on its local structure. Based on this, we develop an algorithm to recognize well-covered graphs. For graphs with n vertices and maximum degree Δ, it runs in linear time if Δ is bounded by a constant, and in polynomial time if $\Delta = O(\sqrt[3]{\log n})$. We mention various applications to areas including hypergraph matchings and radius k independent sets. We extend our results to the problem of determining whether a graph has a weighting which makes it well-covered.

Key words. well-covered graph, maximal independent set, local structure, polynomial time algorithm, recognition algorithm, hypergraph matching, independence system

AMS subject classifications. Primary, 05C75; Secondary, 05B35, 05C70, 05C85, 68R10

PII. S0895480196300479

1. Introduction.

In many situations in mathematics there is a notion of "independence" for subsets of a set. We can "greedily" construct a "large" independent subset by repeatedly adding elements until there are no further feasible elements to add. We often want to know if some property of our final independent set is the same, regardless of how we choose the elements to add. For example, we may wish to determine whether we always get an independent set of maximum size. Or, in a game where players take turns adding elements independent of the previously chosen ones until no further moves can be made, we may ask whether one player inevitably wins. In this paper we show that many natural problems of this kind can be answered by examining their "local structure" and that problems whose local structures satisfy certain size bounds can be solved by polynomial time algorithms. Our results have applications to well-covered graphs, hypergraphs in which maximum or perfect matchings can be found greedily, graphs where certain vertex packings can be found greedily, graph games whose outcome depends on the parity of a maximal independent set, and other related problems.

Our fundamental problem may be formulated in three equivalent ways. In what follows, A is an arbitrary abelian group represented additively, which we use to assign weights to elements of a structure. Our weighted problems all have unweighted

*Received by the editors March 11, 1996; accepted for publication (in revised form) February 2, 1998; published electronically September 1, 1998.
http://www.siam.org/journals/sidma/11-4/30047.html
†Department of Mathematics, School of Education, University of Haifa–Oranim, Tivon, 36006 Israel (zeac603@uvm.haifa.ac.il).
‡Department of Mathematics, 1326 Stevenson Center, Vanderbilt University, Nashville, TN 37240 (mne@math.vanderbilt.edu).
§Mathematics Department, Cumberland College, Williamsburg, KY 40769 (jramey@cc.cumber.edu).

counterparts, which may be considered as the case where $A = \mathbf{Z}$ and all weights are 1; the weight of a set then becomes its cardinality.

Our first formulation is in terms of graphs. An *A-weighted graph* G is a graph whose vertices are weighted by elements of A. Given $S \subseteq V(G)$, the *weight of* S is the sum of the weights of its elements. If all maximal independents sets in G have the same weight, we call it the *independence weight* $\mathrm{iw}(G)$ of G and say that G is *well-covered*.

GRAPH PROBLEM. *Given an A-weighted graph, is it well-covered?*

As multiple edges do not affect the independent sets of vertices and vertices with loops can never appear in an independent set, we assume that our graphs have no loops or multiple edges.

Well-covered (unweighted) graphs were introduced by Plummer [20]. One of the definitions is that a graph is well-covered exactly when all maximal independent sets have the same cardinality, which is just the unweighted version of our more general definition. Plummer [21] recently surveyed known results on well-covered graphs (note that [21, Theorem 5.7] does not correctly summarize the results of [2]). Another special case of the Graph Problem is considered by Finbow and Hartnell [9, 10], who examine the problem of recognizing graphs in which all maximal independent sets have the same parity, i.e., $\mathbf{Z}_2$-weighted graphs with all vertices of weight 1 that are well-covered.

Our second formulation involves independence systems. An *A-weighted independence system* consists of a (finite) set of *points*, a nonempty collection of sets of points known as *independent sets*, which is closed under taking subsets and a function assigning each point a weight in A. The weight of a set of points is the sum of the weights of its elements. A maximal independent set is called a *base*, and a minimal dependent set is called a *circuit*.

INDEPENDENCE SYSTEM PROBLEM. *Given an A-weighted independence system with all circuits of cardinality* 2*, do all bases have the same weight?*

The Independence System Problem is equivalent to the Graph Problem, via the following one-to-one correspondence between independence systems with all circuits of cardinality 2 and graphs. Given a graph, its independent sets of vertices form an independence system, and a circuit is a set containing both ends of any one edge, which has cardinality 2. Conversely, given an independence system with circuits of cardinality 2, we may construct a graph where points become vertices and each circuit becomes the set of ends of an edge. We may also allow circuits of cardinality 1 without essentially changing the problem, as a vertex in a circuit of cardinality 1 can never appear in a base.

Unweighted independence systems with all bases of the same cardinality are the "greedy hereditary systems" of Caro, Sebő, and Tarsi [5], which properly include matroids.

At first glance, considering only independence systems with circuits of cardinality exactly 2 seems very restrictive. However, all circuits are of cardinality 2 when independence of a set of points means pairwise independence of its elements, a very common situation. In particular, the points may be sets, and independence may mean pairwise disjointness. This motivates our third formulation below.

The Independence System Problem expresses our problem in a more abstract way, which allows us easily to recognize many problems as fitting into our framework. For example, Gunther, Hartnell, and Whitehead [15, 16] have considered *radius* 2 *independent sets*, also known as 2-*packings*, in graphs, i.e., sets of vertices which are

pairwise at distance greater than 2. They examine the problem of determining when all maximal radius 2 independent sets have the same cardinality or when they all have the same parity. More generally, Hartnell and Whitehead [18] have examined the cardinality problem for radius k independent sets or k-packings. Clearly these sets form an independence system in which independence means pairwise independence, so our results can provide some information about them. The problems discussed here correspond to $\mathbf{Z}$- and $\mathbf{Z}_2$-weighted independence systems with all weights 1. (This example may also be simply formulated as an instance of the Graph Problem, since a radius k independent set is just an independent set in G^k, the kth power of G.)

Our third formulation involves hypergraphs. An *A-edge-weighted hypergraph* consists of a (finite) set of *vertices*, a collection (with repetitions allowed) of (possibly empty) sets of vertices known as *edges*, and a function giving each edge a weight from A. The weight of a set of edges is the sum of the weights of its elements. A *matching* is a set of mutually disjoint edges.

HYPERGRAPH MATCHING PROBLEM. *Given an A-edge-weighted hypergraph, do all maximal matchings have the same weight?*

The Hypergraph Matching Problem is also equivalent to the Graph Problem, via the following correspondence. Given a graph G, construct a hypergraph H whose vertex set is the edge set of G. For every vertex v of G, the set of edges incident with v gives an edge e_v of H; the weight of e_v is just the weight of v. Conversely, given a hypergraph H, construct a graph G whose vertex set is the edge set of H with two vertices of G adjacent if the corresponding edges of H intersect; the weight of a vertex in G is just the weight of the corresponding edge in H.

Many aspects of the unweighted version of the Hypergraph Matching Problem are discussed in [5, Section 2], including applications to two graph decomposition problems studied by Ruiz [24] and Caro, Ruiz, and Rojas [4]. As an example of the use of weights in the Hypergraph Matching Problem, suppose we assign each edge a weight from $\mathbf{Z}$ equal to its cardinality. Then our question is equivalent to asking whether all maximal matchings use the same number of vertices. In particular, we may ask if they are all *perfect*, i.e., use all the vertices.

Since all three problems above are equivalent, we may work with whichever is most convenient for a particular application. For the purposes of this paper it is easiest to work with the Graph Problem. The following is known about the computational complexity of this problem. In the unweighted case, recognition of well-covered graphs is a co-NP-complete problem, as shown by Chvátal and Slater [8] and, independently, Sankaranarayana and Stewart [25]. It is co-NP-complete even when a graph has no induced $K_{1,4}$'s [5, Theorem 2.1], although polynomial algorithms have been found for line graphs [19] and claw-free graphs—graphs with no induced $K_{1,3}$ [26, 27]. Moreover, structural characterizations of certain classes of well-covered graphs easily yield recognition algorithms, such as for cubic graphs [2] and for graphs of girth at least 5 [11] (the *girth* is the length of a shortest cycle in the graph). Other structural work on well-covered graphs includes work by Ravindra [23] on bipartite graphs, Ramey on graphs of maximum degree at most 3 [22], and by various authors on graphs with no 4-cycles [12, 14, 17].

If a fixed abelian group A can be represented by finite strings that can be added in polynomial time, then the graph problem, i.e., the recognition problem for well-covered A-weighted graphs, is co-NP-complete, even for $K_{1,4}$-free graphs: the unweighted proof of [5, Theorem 2.1] is easily modified by giving each vertex the same nonzero weight. In particular, the problem of recognizing graphs in which all maximal

independent sets have the same parity, as studied by Finbow and Hartnell [9, 10] is co-NP-complete for $K_{1,4}$-free graphs.

We will examine minimal non-well-covered graphs, which arise in characterizing well-covered graphs, and use them to give an algorithm for recognizing well-covered graphs. For n-vertex graphs this runs in polynomial time if the maximum degree is bounded by $O(\sqrt[3]{\log n})$ and in linear time if the maximum degree is bounded by a constant. This answers a question posed in [2], as to whether well-covered graphs of bounded degree can be recognized in polynomial time.

2. Minimal non-well-covered graphs. In this section we discuss ways to characterize well-covered A-weighted graphs, leading to the notion of a minimal non-well-covered graph. When no confusion can result, we just refer to an A-weighted graph as a *graph*. Unless explicitly stated otherwise, vertices in subgraphs of a graph G inherit their weights from G, and when we combine graphs $G_1, G_2, \ldots, G_k$ to obtain G, in such a way that $V(G)$ is the disjoint union of $V(G_1), V(G_2), \ldots, V(G_k)$, then each vertex of G inherits its weight from the appropriate G_i.

If S is a set of vertices in a graph G, the *closed neighborhood* of S is the set $N_G[S]$, or just $N[S]$, containing S and all neighbors of vertices in S. We abbreviate $N[\{v\}]$ to $N[v]$. If S is independent, the graph $G\backslash N[S]$ is said to be obtained from G by *neighborhood deletion*. The following important observation seems to have been first stated by Campbell [1] for unweighted graphs. A similar result in the special context of randomly decomposable graphs was observed by Caro, Rojas, and Ruiz [4].

OBSERVATION 2.1. *Suppose $S \subseteq V(G)$ is independent. If G is well-covered, then $G\backslash N[S]$ is well-covered. Equivalently, if any component of $G\backslash N[S]$ is not well-covered, then G is not well-covered.*

This observation is very useful in characterizing classes of well-covered graphs. If we can show that a certain structure in a graph G contains an independent set S of G such that $G\backslash N[S]$ has a component which is a non-well-covered graph L, then that structure cannot occur in a well-covered graph. We are, therefore, interested in generating (isomorphism classes of) non-well-covered graphs L, which can be used to restrict the possible structure of well-covered graphs.

Some non-well-covered graphs L are not needed to characterize well-covered graphs, because any structures they eliminate can also be eliminated by smaller non-well-covered graphs. In particular, suppose that L contains a nonempty independent set T for which $M = L\backslash N_L[T]$ is non-well-covered. Then, whenever S is an independent set in G such that $G\backslash N[S]$ contains L as a component, $S \cup T$ is also an independent set in G such that $G\backslash(S \cup T)$ contains M as a component. In other words, M can eliminate any structures that L can. The essential non-well-covered graphs L, those that cannot be replaced in this way, are those for which $L\backslash N_L[T]$ is well-covered for all nonempty independent T in L. Thus, they are the non-well-covered graphs which are minimal with respect to the neighborhood deletion operation; we call them simply *minimal non-well-covered graphs*. In the unweighted case, translated into the hypergraph matching form of our problem, they correspond exactly to the *critical nongreedy hypergraphs* investigated by Caro, Sebő, and Tarsi [5, Section 2.7], and they generalize an idea developed by Caro, Rojas, and Ruiz [4] in the context of randomly decomposable graphs. We may summarize the usefulness of minimal non-well-covered graphs as follows.

OBSERVATION 2.2. *G is non-well-covered if and only if there exists some (possibly empty) independent set S in G such that $G\backslash N[S]$ has a component which is minimal non-well-covered.*

We now characterize minimal non-well-covered graphs. Our characterization is a natural generalization of the characterizations for the unweighted case obtained independently by Ramey [22, Theorems 2.11, 2.12] and by Caro, Sebő, and Tarsi (using the hypergraph matching form of the problem) [5, Theorem 2.8]. Our proof adapts that of [5]. Note that $G_1 + G_2 + \cdots + G_k$ denotes the *join* of G_1, G_2, ..., G_k, obtained from their disjoint union by adding an edge between every pair of vertices in different graphs.

THEOREM 2.3. *An A-weighted graph G is minimal non-well-covered if and only if there exist well-covered A-weighted graphs $G_1, G_2, \ldots, G_k$ such that $G = G_1 + G_2 + \cdots + G_k$ and $\mathrm{iw}(G_i) \neq \mathrm{iw}(G_j)$ for some i and j.*

Proof. Suppose G is minimal non-well-covered. For each $v \in V(G)$, the graph $G \backslash N[v]$ is well-covered, and therefore every maximal independent set of G containing v has the same weight, which we denote $t(v)$. Let $\{t(v) : v \in V(G)\}$ be $\{t_1, t_2, \ldots, t_k\}$; since G is not well-covered, $k \geq 2$. Let G_i be the subgraph of G induced by $\{v \in V(G) : t(v) = t_i\}$. If $v \in V(G_i)$ and $w \in V(G_j)$, then v and w must be adjacent, for otherwise there would be a maximal independent set including $\{v, w\}$, and we would have $t(v) = t(w)$. Thus, $G = G_1 + G_2 + \cdots + G_k$, and a set of vertices of G is independent if and only if it is an independent set in some G_i. Therefore, any maximal independent set in G_i is a maximal independent set in G and has weight t_i, so that G_i is well-covered with $\mathrm{iw}(G_i) = t_i$. Since $k \geq 2$, $\mathrm{iw}(G_i) \neq \mathrm{iw}(G_j)$ for some i and j.

Suppose now that $G = G_1 + G_2 + \cdots + G_k$, where $G_1, G_2, \ldots, G_k$ are all well-covered. Clearly, any set $S \subseteq V(G)$ is independent if and only if S is an independent set in G_i for some i. Therefore, the maximal independent sets of G are the maximal independent sets of the individual G_i's, and G is well-covered if and only if $\mathrm{iw}(G_i)$ is the same for all i. Thus, if $\mathrm{iw}(G_i) \neq \mathrm{iw}(G_j)$ for some i and j, then G is non-well-covered. Moreover, it is minimal with respect to neighborhood deletion, because for any nonempty independent S in G_i, we have $G \backslash N_G[S] = G_i \backslash N_{G_i}[S]$, which is well-covered by Observation 2.1. □

The following corollary will be very important in the next section.

COROLLARY 2.4. *A minimal non-well-covered A-weighted graph has diameter at most 2.*

Proof. Such a graph is a join, and any join has diameter at most 2. □

Some special cases of Theorem 2.3 are of interest. Suppose G is unweighted and minimal non-well-covered. If G is bipartite, or, in fact, if G has girth 4 or more, then G must be a complete bipartite graph $K_{m,n}$ with $1 \leq m < n$. If G has girth 5 or more, then G must be a star $K_{1,n}$ with $n \geq 2$. The fact that the minimal non-well-covered graphs in these situations can easily be described seems to be reflected in the fact that well-covered unweighted graphs that are bipartite or have girth 5 or more have relatively simple characterizations [11, 23]. It also suggests that the problem of characterizing well-covered graphs of girth 4, which has been considered difficult, may in fact be tractable. Note also that a consequence of the unweighted version of Theorem 2.3 has been used by Tankus and Tarsi [27] to find a simple proof of their earlier result [26] that well-covered claw-free graphs can be recognized in polynomial time.

Theorem 2.3 also has some implications for the characterization of well-covered graphs of bounded degree.

COROLLARY 2.5. *Let G be a minimal non-well-covered A-weighted graph. If G has maximum degree Δ, then G contains at most 2Δ vertices.*

Proof. By Theorem 2.3, we know that $V(G)$ can be partitioned into two sets (e.g., $V(G_1)$ and $V(G_2) \cup V(G_3) \cup \cdots \cup V(G_k)$) such that every possible edge between the sets is in $E(G)$. Since the maximum degree is Δ, each set cannot contain more than Δ vertices; otherwise, the vertices in the other set would have degree greater than Δ. □

In the unweighted case, this corollary is close to the best possible, as shown by the complete bipartite graph $K_{\Delta-1,\Delta}$, which is minimal non-well-covered on $2\Delta - 1$ vertices with maximum degree Δ. It is best possible in situations where $K_{\Delta,\Delta}$ with bipartition (V_1, V_2) can be assigned weights so that V_1 and V_2 have different weights.

Corollary 2.5 implies immediately that in the unweighted case, or if A is finite, there are finitely many minimal non-well-covered graphs of maximum degree at most Δ. All may be constructed as joins of well-covered graphs, not all with the same independence weight, on at most Δ vertices. We summarize some results for the unweighted case. There is only one minimal non-well-covered graph with maximum degree 2, namely, $P_3 = K_{1,2} = K_1 + 2K_1$. The four minimal graphs with maximum degree 3 are $K_{1,3} = K_1 + 3K_1$, $K_1 + (K_2 \cup K_1)$, $K_{1,1,2} = K_1 + K_1 + 2K_1$, and $K_{2,3} = 2K_1 + 3K_1$. There are 14 minimal graphs with maximum degree 4 and 43 minimal graphs with maximum degree 5. The minimal graphs with maximum degree at most 3 were used, without realizing their nature, in [2] in characterizing well-covered cubic graphs, and were consciously used by Ramey [22] to characterize the well-covered graphs with maximum degree at most 3.

3. Testing well-coveredness. In this section, we use Corollaries 2.4 and 2.5 to prove that well-coveredness depends on the local, rather than the global, structure of a graph. We then show how this can be used to test whether a graph is well-covered, resulting in polynomial time algorithms under certain circumstances. Let $N_k[v]$ denote the set of vertices at distance at most k from a vertex v in the graph G.

THEOREM 3.1. *Consider an A-weighted graph G with maximum degree Δ. The following are equivalent.*

(i) *G is non-well-covered.*

(ii) *There exist $v \in V(G)$ and an independent set S in the subgraph $Q = Q(v)$ of G induced by $N_4[v]$, such that $Q \backslash N_Q[S]$ has a minimal non-well-covered component containing v.*

(iii) *There exist $v \in V(G)$ and an independent set S in the subgraph $Q = Q(v)$ of G induced by $N_4[v]$, such that $Q \backslash N_Q[S]$ has a non-well-covered component with at most 2Δ vertices and of diameter at most 2 containing v.*

Proof. (i) $\Rightarrow$ (ii): If G is non-well-covered, by Observation 2.2 we can choose an independent set S, minimal with respect to inclusion, for which $G \backslash N[S]$ has a minimal non-well-covered component L. A vertex of S cannot be at distance 0 or 1 from L, and if it is at distance 3 or more from L then we may delete it from S without changing the fact that L is a component of $G \backslash N[S]$. Thus, by minimality of S, every vertex of S is at distance 2 from L. Let v be an arbitrary vertex of L. Since L has diameter at most 2 by Corollary 2.4, every vertex of S is in $N_4[v]$ and L is a component of $Q \backslash N_Q[S]$, where Q is induced by $N_4[v]$.

(ii) $\Rightarrow$ (iii): By Corollaries 2.4 and 2.5, a minimal non-well-covered component has the properties specified in (iii).

(iii) $\Rightarrow$ (i): If v, S, and Q exist as stated in (iii), let L be the non-well-covered component of $Q \backslash N_Q[S]$ containing v. Since L has diameter at most 2, no vertex of L is adjacent in G to a vertex outside Q. Therefore, L is also a component of $G \backslash N_G[S]$ and G is non-well-covered. □

For creating theoretical characterizations of classes of well-covered graphs, condition (ii) is very useful. Condition (iii) is easier to check by computer than (ii), so it is useful in constructing algorithms.

THEOREM 3.2. *Let G be an A-weighted graph with n vertices and maximum degree Δ, represented by a list of neighbors for each vertex. Then we may determine whether or not G is well-covered in $O(ne^{2\Delta}\Delta^{2\Delta+5/2}2^{2\Delta^3-4\Delta^2})$ operations (or, more roughly, in $O(n2^{2\Delta^3})$ operations). (Each addition or comparison in A is counted as one operation.)*

Proof. We check condition (iii) of Theorem 3.1 by brute force. Given a vertex v, we try to find a non-well-covered component L, containing v, of some $Q\backslash N_Q[S]$, by first constructing all possible graphs L and then trying to find S.

Fix v. There are at most $\Delta + \Delta(\Delta-1) = \Delta^2$ vertices at distance 1 or 2 from v, which we locate in $O(\Delta^2)$ operations. The number of sets of vertices of cardinality at most 2Δ that include v and otherwise contain only vertices at distance 1 or 2 from v is at most

$$\binom{\Delta^2}{0} + \binom{\Delta^2}{1} + \cdots + \binom{\Delta^2}{2\Delta-1} = O(e^{2\Delta}2^{-2\Delta}\Delta^{2\Delta-3/2})$$

(using Stirling's formula and other standard estimations). The subgraph L induced by such a set has at most 2Δ vertices and maximum degree at most Δ. Thus, L may be generated in $O(\Delta^2)$ operations (including the generation of its vertex set), and checked for diameter 2 in $O(\Delta^3)$ operations. Each of the $O(2^{2\Delta})$ subsets of $V(L)$ may be generated, be checked to see if it is a maximal independent set in L, and have its weight calculated in $O(\Delta^2)$ operations, so we require $O(\Delta^2 2^{2\Delta})$ operations to determine whether L is non-well-covered.

Now, given a non-well-covered diameter 2 graph L, we must see if S exists with $Q\backslash N_Q[S]$ having L as a component. As in the proof of Theorem 3.1, we may assume that every vertex of S is at distance 2 from L. Given a set S of vertices at distance 2 from L, we need check only that S is independent and that every vertex at distance 1 from L is covered by, i.e., adjacent to, a vertex of S. Since L has at most 2Δ vertices, there are at most $2\Delta(\Delta-1)$ vertices at distance 1 from L, and at most $2\Delta(\Delta-1)^2$ vertices at distance 2 from L. Therefore, there are $O(2^{2\Delta(\Delta-1)^2})$ potential sets S, each of which can be generated in $O(\Delta^3)$ operations and checked for independence and covering in $O(\Delta^4)$ operations.

Combining the above estimates, we come up with a bound on the number of operations in which the dominant term comes from the number of vertices v, the number of sets giving a possible L, the number of possible sets S, and the time to check each S for independence and covering. Thus the total number of operations required is

$$O(n \cdot e^{2\Delta}2^{-2\Delta}\Delta^{2\Delta-3/2} \cdot 2^{2\Delta(\Delta-1)^2} \cdot \Delta^4) = O(ne^{2\Delta}\Delta^{2\Delta+5/2}2^{2\Delta^3-4\Delta^2}),$$

which is clearly $O(n2^{2\Delta^3})$. ∎

In some cases, where the maximum degree is large but there are only a few vertices of maximum degree and they are widely separated, it may be more useful to observe that the above method also uses at most $O(nq^2 2^{2q})$ operations, where q is the maximum of $|N_4[v]|$ for $v \in V(G)$. For example, if most vertices have degree 3 or less, with any two vertices of degree 4 or more always being separated by distance 5 or more, then $q \le 15\Delta + 1$, and so we obtain a bound on the number of operations of $O(n\Delta^2 2^{30\Delta})$, which is better than a bound involving $2^{2\Delta^3}$.

Theorem 3.1, in fact, implies that in a non-well-covered graph G, there is some vertex for which the subgraph $Q = Q(v)$ induced by $N_4[v]$ is non-well-covered. If the converse of this statement were true, it would give a very simple local characterization of well-covered graphs. Unfortunately, the converse is false, as shown by the graph G obtained from a 9-cycle by joining one pendant vertex to each original vertex: this graph is well-covered, but every $Q(v)$ is non-well-covered.

We can now answer the question posed in [2], of whether well-covered graphs of bounded degree can be recognized in polynomial time.

COROLLARY 3.3. *Suppose that addition and comparison in A can be done in polynomial time. For graphs, let n denote the number of vertices and Δ the maximum degree.*

(i) *Suppose $\mathcal{F}$ is a family of A-weighted graphs such that $\Delta = O(\sqrt[3]{\log n})$. Then we may determine whether a graph in $\mathcal{F}$ is well-covered in polynomial time.*

(ii) *Suppose $\mathcal{F}$ is a family of A-weighted graphs such that $\Delta = O(1)$, i.e., Δ is bounded by a constant. Then we may determine whether a graph in $\mathcal{F}$ is well-covered in linear time ($O(n)$ operations).*

We now mention some applications of the above, to some of the specific problems mentioned earlier and to some classes of graphs derived from (unweighted) well-covered graphs. We expect that our results will also apply to many other interesting situations involving "greedy" or "random" processes.

COROLLARY 3.4. *The following questions may be settled in polynomial time:*

(a) *Given a family of graphs with $\Delta = O(\sqrt[3]{\log n})$, do all maximal independent sets have the same parity*?

(b) *Given a family of graphs with $\Delta = O(\sqrt[3k]{\log n})$, do all radius k independence sets (k-packings) have the same cardinality*? *Or do they all have the same parity*?

(c) *Given a family of hypergraphs where the number of edges is m and each edge intersects at most $O(\sqrt[3]{\log m})$ other edges, do all matchings have the same cardinality*? *Or are all matchings perfect*?

(d) *Given a family of graphs with $\Delta = O(\sqrt[3]{\log n})$, is a graph in the class W_2 (also known as 1-well-covered) or is it strongly well-covered*? (*See* [21] *for definitions and work on these concepts. One of us (Caro) has shown that the first of these two problems is co-NP-complete for general graphs or even $K_{1,4}$-free graphs* [3].)

Is it possible to significantly improve Theorem 3.2? For example, is there $\epsilon > 0$ such that determining whether a graph (weighted or unweighted) is well-covered can be done in polynomial time when $\Delta = O(n^\epsilon)$? For any $\epsilon > 0$, this problem is co-NP-complete, because for any graph G with n vertices, determining whether G is well-covered is equivalent to determining whether the union of $n^{1/\epsilon - 1}$ disjoint copies of G is well-covered and for this union, $\Delta = O(n^\epsilon)$.

Theorem 3.2 and the previous paragraph refer to graphs which are sparse. Some observations may also be made for dense graphs. Let $\overline{\Delta}$ denote the maximum degree of the complement of a graph, equal to $n - 1 - \delta$, where δ is the minimum degree. If $\overline{\Delta} = O(\log n)$, then any vertex has at most $O(\log n)$ nonneighbors, and all independent sets may be constructed and tested for maximality in polynomial time, so determining whether a graph (weighted or unweighted) is well-covered may be done in polynomial time. However, for any $\epsilon > 0$, the problem of determining whether graphs with $\overline{\Delta} = O(n^\epsilon)$ are well-covered is co-NP-complete. For any G, determining whether G is well-covered is equivalent to determining whether the join of $n^{1/\epsilon - 1}$ disjoint copies of G is well-covered and for this join, $\overline{\Delta} = O(n^\epsilon)$.

Algorithmically, it may be possible to take advantage of the ease of finding a

maximum independent set in an unweighted well-covered graph, without explicitly being able to recognize well-covered graphs. Jerry Spinrad (personal communication) has posed the following problem: Is there a polynomial time algorithm which, given any graph, either finds a maximum independent set for the graph or reports that the graph is not well-covered? A weighted version of this makes sense only in the context of nonnegative real weights.

Finally, it is interesting to contrast the behavior of two closely-related problems: determining whether a graph is well-covered and finding a maximum independent set. The recognition problem for well-covered graphs is co-NP-complete [8, 25], but for any constant Δ, we can recognize well-covered graphs of maximum degree at most Δ in linear time. The maximum independent set problem, however, remains NP-complete even for cubic planar graphs (see [13, p. 194] for references).

4. Well-covered weightings. In this section we consider the following concept. Given an abelian group A and an unweighted graph G, a function $x : V(G) \to A$ is called a *well-covered weighting* of G if it makes G into a well-covered A-weighted graph. The zero function is always a well-covered weighting, but does a graph have any nonzero (i.e., nonzero for at least one vertex) well-covered weighting? Sometimes we may wish to add stronger restrictions, such as that the weighting must be nonzero for *all* vertices, or that it must take nonnegative or positive values if A is an additive subgroup of $\mathbf{R}$.

Not every graph has a nonzero well-covered weighting. To give two arbitrary examples, the Petersen graph and every cycle of length 8 or more have no nonzero well-covered weighting over any abelian group. It is not difficult to prove this by using Observation 2.1 to derive properties of a well-covered weighting and then deduce that it must be zero.

Let $\mathcal{B}(G)$ denote the set of maximal independent sets, i.e., bases, of G. If x is a well-covered weighting, then $x(B)$ must be the same for all $B \in \mathcal{B}(G)$, i.e., if B_0 is a fixed element of $\mathcal{B}(G)$, then

$$x(B) - x(B_0) = 0 \quad \forall B \in \mathcal{B}(G) \backslash \{B_0\}.$$

We call this system of equations in the variables $x(v)$, $v \in V(G)$, a *global well-covering system* for G. When A is a commutative ring with identity, as well as an additive abelian group, it is a linear system over A. It is a finite system, but, in general, it will have exponentially many equations, and so we will not be able to determine in polynomial time if there is a nonzero well-covered weighting of G.

However, we can replace a global system by another system using Theorem 3.1. In the following discussion, we use the notation from that theorem. For each $v \in V(G)$, let $\mathcal{L}(v)$ denote the set of all subgraphs L which (i) are obtained as a component of $Q(v) \backslash N_{Q(v)}[S]$ for some independent S in $Q(v)$, (ii) contain v, (iii) contain at most $2\Delta(G)$ vertices, and (iv) have diameter at most 2. By Theorem 3.1, G is well-covered if and only if each element of $\mathcal{L}(G) = \cup_{v \in V(G)} \mathcal{L}(v)$ is well-covered. Therefore, a weight function makes G into a well-covered graph if and only if it makes each element of $\mathcal{L}(G)$ well-covered. Thus, the well-covering system for G has the same solution set as the union of a global well-covering system for each element of $\mathcal{L}(G)$: we will call this union a *local well-covering system* for G. Constructing a local well-covering system requires only a minor modification in the algorithm of Theorem 3.2. Instead of a step to compute and compare its weight for a maximal independent set in a graph $L \in \mathcal{L}(G)$, we have a step to set up its equation. Therefore, a local well-covering system can be constructed in $O(n2^{2\Delta^3})$ operations. In particular, if $\Delta = O(\sqrt[3]{\log n})$,

we obtain a polynomial time algorithm to set up a local well-covering system. By Gaussian elimination, we can then find a basis for the solutions of that system in polynomial time, and so we have a polynomial time algorithm to determine a basis for the space of well-covered weightings of G.

As a minor modification of the above, a local well-covering system may be used to compute the rank over a given field of the maximal independent set incidence matrix X of a graph (rows are indexed by maximal independent sets S, columns by vertices v, with an entry being 1 if $v \in S$ and 0 otherwise). Since a local well-covering system is equivalent to any global well-covering system, the rows of the matrix of a local well-covering system together with the incidence vector of any one maximal independent set span precisely the rowspace of X. If $\Delta = O(\sqrt[3]{\log n})$ this gives a polynomial time algorithm for finding the rank of X. In the case where the graph is (unweighted) well-covered, every row of X has the same number of 1's and so X has some special structure; however, the algorithm is valid regardless of whether the graph is well-covered, which is a little surprising.

A case which appears to have special interest is the case of positive real, rational, or integral weightings. Working with real or rational numbers is essentially the same, as in either case we have a basis for the solution set of a (global or local) well-covering system using only rational numbers, because all coefficients in the system are integral (in fact, 0 or ± 1). And working with rational numbers or integers is essentially the same, because to obtain an integral solution we merely multiply by a constant to clear the denominator in a rational solution. Therefore, we assume that we are working with rational numbers. Not all graphs have a positive well-covered weighting. The simplest example known to us is P_5; the central vertex must have weight 0 in any well-covered weighting over any A. We would guess that determining whether a positive well-covered weighting exists is, in general, a difficult problem.

If we have a positive well-covered weighting for a graph, then by multiplying by a suitable positive constant, we know that there is a positive well-covered weighting for which all weights are at least 1. Therefore, we can formulate the question as to whether a graph has a positive well-covered weighting as follows: Is there a weighting which satisfies a (global or local) well-covering system, and for which all weights are at least 1? This feasibility problem is solvable via linear programming, and thus there is a polynomial algorithm to solve it when a well-covering system can be found in polynomial time, e.g., when $\Delta = O(\sqrt[3]{\log n})$. The existence problem for nonnegative nonzero well-covered weightings can be solved in a similar way.

As a final remark, note that the idea of the space of well-covered weightings (vertex weightings with a uniform sum on the maximal independent sets of a graph) has been extended by Caro and Yuster [6, 7] to the idea of a *uniformity space* (the vertex weightings with a uniform sum on the edges of an arbitrary hypergraph).

REFERENCES

[1] S. R. Campbell, *Some Results on Cubic Well-Covered Graphs*, Ph.D. thesis, Vanderbilt University, Nashville, TN, 1987.

[2] S. R. Campbell, M. N. Ellingham, and G. F. Royle, *A characterisation of well-covered cubic graphs*, J. Combin. Math. Combin. Comput., 13 (1993), pp. 193–212.

[3] Y. Caro, *Subdivisions, parity and well-covered graphs*, J. Graph Theory, 25 (1997), pp. 85–94.

[4] Y. Caro, J. Rojas, and S. Ruiz, *A forbidden subgraphs characterization and a polynomial algorithm for randomly decomposable graphs*, Czechoslovak Math. J., 46 (1996), pp. 413–419.

[5] Y. Caro, A. Sebő, and M. Tarsi, *Recognizing greedy structures*, J. Algorithms, 20 (1996), pp. 137–156.
[6] Y. Caro and R. Yuster, *The Uniformity Space of Graphs and Hypergraphs and Its Applications*, preprint, University of Haifa-ORANIM, Tivon, Israel, 1996.
[7] Y. Caro and R. Yuster, *The Zero-Sum Mod* 2 *Bipartite Ramsey Numbers and the Uniformity Space of Bipartite Graphs*, preprint, University of Haifa-ORANIM, Tivon, Israel, 1996.
[8] V. Chvátal and P. J. Slater, *A note on well-covered graphs*, in Quo Vadis, Graph Theory?, Ann. Discrete Math. 55, North–Holland, Amsterdam, 1993, pp. 179–182.
[9] A. Finbow and B. L. Hartnell, *A game related to covering by stars*, Ars Combin., 16-A (1983), pp. 189–198.
[10] A. Finbow and B. Hartnell, *A characterization of parity graphs containing no cycle of order five or less*, Ars Combin., 40 (1995), pp. 227–234.
[11] A. Finbow, B. Hartnell, and R. Nowakowski, *A characterization of well-covered graphs of girth* 5 *or greater*, J. Combin. Theory Ser. B, 57 (1993), pp. 44–68.
[12] A. Finbow, B. Hartnell, and R. Nowakowski, *A characterization of well-covered graphs that contain neither* 4- *nor* 5*-cycles*, J. Graph Theory, 18 (1994), pp. 713–721.
[13] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, CA, 1979.
[14] S. L. Gasquoine, B. Hartnell, R. Nowakowski, and C. Whitehead, *Techniques for constructing well-covered graphs with no* 4*-cycles*, J. Combin. Math. Combin. Comput., 17 (1995), pp. 65–87.
[15] G. Gunther, B. Hartnell, and C. A. Whitehead, *On* 2*-packings of graphs of girth at least* 9, in Proc. 26th Southeastern Internat. Conf. on Combinatorics, Graph Theory, and Computing, Boca Raton, FL, Congr. Numer., 110 (1995), pp. 211–222.
[16] B. L. Hartnell, *On maximal radius* 2 *independent sets*, Congr. Numer., 48 (1985), pp. 179–182.
[17] B. L. Hartnell, *On the local structure of well-covered graphs without* 4*-cycles*, Ars Combin., 45 (1997), pp. 77–86.
[18] B. Hartnell and C. A. Whitehead, *On* k*-Packings of Graphs*, preprint, St. Mary's University, Halifax, NS, Canada, 1995.
[19] M. Lesk, M. D. Plummer, and W. R. Pulleyblank, *Equimatchable graphs*, in Graph Theory and Combinatorics, B. Bollobás, ed., Academic Press, London, 1984, pp. 239–254.
[20] M. D. Plummer, *Some covering concepts in graphs*, J. Combin. Theory, 8 (1970), pp. 91–98.
[21] M. D. Plummer, *Well-covered graphs: A survey*, Quaestiones Math., 16 (1993), pp. 253–287.
[22] J. E. Ramey, *Well-Covered Graphs with Maximum Degree Three and Minimal Non-Well-Covered Graphs*, Ph.D. thesis, Vanderbilt University, Nashville, TN, 1994.
[23] G. Ravindra, *Well-covered graphs*, J. Combin. Inform. System Sci., 2 (1977), pp. 20–21.
[24] S. Ruiz, *Randomly decomposable graphs*, Discrete Math., 57 (1985), pp. 123–128.
[25] R. S. Sankaranarayana and L. K. Stewart, *Complexity results for well-covered graphs*, Networks, 22 (1992), pp. 247–262.
[26] D. Tankus and M. Tarsi, *Well-covered claw-free graphs*, J. Combin. Theory Ser. B, 66 (1996), pp. 293–302.
[27] D. Tankus and M. Tarsi, *The structure of well-covered graphs and the complexity of their recognition problems*, J. Combin. Theory Ser. B, 69 (1997), pp. 230–233.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 655–663, November 1998

010

ON THE WEAKNESS OF AN ORDERED SET*

JOHN G. GIMBEL† AND ANN N. TRENK‡

Abstract. In this paper we extend the notion of a ranking of elements in a weak order to a ranking of elements in general ordered sets. The *weakness* of an ordered set $P = (X, \prec)$ (denoted $wk(P)$) is the minimum integer k for which there exists an integer-valued function $lev : X \to \mathbf{Z}$ satisfying: (i) if $x \prec y$, then $lev(x) < lev(y)$; and (ii) if $x \parallel y$, then $|lev(x) - lev(y)| \leq k$ (where "$\parallel$" denotes incomparability). A *forcing cycle* L in P is a sequence of elements $L : x = v_0, v_1, \ldots, v_m = x$ of P so that for each $i \in \{0, 1, \ldots, m-1\}$ either $v_i \prec v_{i+1}$ or $v_i \parallel v_{i+1}$.

Our main result relates these two concepts; we prove $wk(P) = \max_L \left\lceil \frac{up(L)}{side(L)} \right\rceil$, where $up(L) = \#\{i : v_i \prec v_{i+1}\}$, $side(L) = \#\{i : v_i \parallel v_{i+1}\}$ and the maximum is taken over all forcing cycles L in P.

We also discuss algorithms for computing $wk(P)$ and prove that $wk(P)$ is a comparability invariant.

Key words. partially ordered sets, weak orders

AMS subject classifications. 06A06, 68R10

PII. S0895480197319628

1. Introduction.

We begin with some notation. The ordered sets in this paper will be irreflexive, with "$\prec$" denoting the relation, unless otherwise specified. If x and y are incomparable elements, we write $x \parallel y$. We denote by $\underline{r} + \underline{s}$ the ordered set consisting of two disjoint chains, one with r elements, the other with s elements (the ordered set $\underline{3} + \underline{4}$ is shown in Figure 2.2).

DEFINITION 1.1. *Given an ordered set $P = (X, \prec)$, and a nonnegative integer k, an integer-valued function $lev : X \to \mathbf{Z}$ is a* k-leveling function *of P if it satisfies the following for all $x, y \in X$.*

Rule A. *If $x \prec y$, then $lev(x) < lev(y)$.*

Rule B. *If $x \parallel y$, then $|lev(x) - lev(y)| \leq k$.*

An ordered set $P = (X, \prec)$ is *k-weak* if there exists a k-leveling function of P. The *weakness* of an ordered set P, denoted $wk(P)$, is the least k for which P is k-weak.

It is easy to see that an ordered set P with n elements has $wk(P) \leq n-1$ simply by taking a linear extension of P and assigning $lev(x)$ to be the height of x in the linear extension. Thus the concept of weakness is well defined. In Proposition 3.1 and Theorem 3.4 we improve this bound to $wk(P) \leq \min(hgt(P), \lceil \frac{n-2}{2} \rceil)$, where $hgt(P)$ is the number of elements in a maximum chain in P.

In the case $k = 0$, Rules A and B can be combined as follows: A 0-leveling function $lev : X \to \mathbf{Z}$ is one that satisfies $x \prec y$ iff $lev(x) < lev(y)$ for all $x, y \in X$. The resulting 0-weak orders are known as *weak orders* [1]. Thus there is a natural ranking of the elements of a 0-weak order, which provides an ordering that is almost linear, except that ties are allowed. One example of a weak ordering is a set of students ordered by grade point average.

*Received by the editors April 8, 1997; accepted for publication (in revised form) March 31, 1998; published electronically September 1, 1998.

http://www.siam.org/journals/sidma/11-4/31962.html

†Department of Mathematical Sciences, University of Alaska, Fairbanks, AK 99775-1110 (ffjgg@uaf.edu).

‡Department of Mathematics, Wellesley College, Wellesley, MA 02181 (atrenk@wellesley.edu).
The research of this author was supported in part by DIMACS.

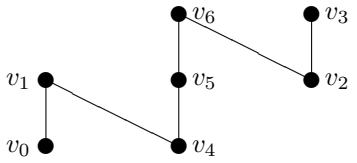


FIG. 2.1. *An ordered set Q with $wk(Q) = 2$.*

More generally, there may be several criteria for ordering elements. As an example, the management of a company may want to order employees based on performance. Such an order is unlikely to be a weak order. Yet it is still desirable to assign a rank (level) to each employee for the purposes of salary computation. If employee y is superior to employee x, y should get a higher salary than x (Rule A). In addition, it is desirable to minimize the largest salary discrepancy between incomparable employees. This is achieved by using a k-leveling function where k is as small as possible, i.e., $k = wk(P)$.

The rest of the paper is organized as follows. In section 2 we introduce forcing cycles and prove our main theorem which relates weakness to forcing cycles. We prove two upper bounds on weakness in section 3 and consider algorithms to compute weakness in section 4. Finally, in section 5 we prove that weakness is a comparability invariant.

2. Forcing cycles. In this section we characterize the weakness of an ordered set in terms of what we call forcing cycles. Each forcing cycle in P provides a lower bound on $wk(P)$. Surprisingly, $wk(P)$ is completely determined by the forcing cycles in P.

Given an ordered set P, a *forcing cycle* L in P is a sequence of elements $L : x = v_0, v_1, \ldots, v_m = x$ of P so that for each $i \in \{0, 1, \ldots, m-1\}$ either $v_i \prec v_{i+1}$ or $v_i \parallel v_{i+1}$. The element x is called the *starting point* of cycle L. Given a forcing cycle $L : x = v_0, v_1, \ldots, v_m = x$ of an ordered set, we define $up(L) = \#\{i : v_i \prec v_{i+1}\}$ and $side(L) = \#\{i : v_i \parallel v_{i+1}\}$. Thus $up(L) + side(L) = m$. The sequence $L' : v_0, v_1, \ldots, v_6, v_0$ is a forcing cycle in the ordered set shown in Figure 2.1. Note that $up(L') = 4$ and $side(L') = 3$.

We make a few remarks about forcing cycles before stating the main result of this paper.

1. Any element may be chosen as the starting point of a forcing cycle. This is true because if $L : x = v_0, v_1, \ldots, v_m = x$ is a forcing cycle, so is $L_i : v_i, v_{i+1}, \ldots, v_m = v_0, v_1, \ldots, v_i$ for each $i < m$.
2. The quantities $up(L)$ and $side(L)$ are independent of the starting point of L.
3. Every forcing cycle L has $side(L) \geq 2$.
   Proof of 3. Let $L : x = v_0, v_1, \ldots, v_m = x$ be a forcing cycle of the ordered set $P = (X, \prec)$. If $side(L) = 0$, then $x \prec v_1 \prec \cdots \prec v_{m-1} \prec x$ which implies $x \prec x$, contradicting the irreflexivity of P. If $side(L) = 1$, then there exists $i \in \{1, 2, \ldots, m-1\}$ such that $x = v_0 \prec v_1 \prec \cdots \prec v_i$, $v_i \parallel v_{i+1}$, and $v_{i+1} \prec v_{i+2} \prec \cdots \prec v_m = x$. But then by transitivity, $v_{i+1} \prec x \prec v_i$, and therefore, $v_{i+1} \prec v_i$, which contradicts $v_i \parallel v_{i+1}$. Thus $side(L) \geq 2$.

THEOREM 2.1. *If $P = (X, \prec)$ is an ordered set, then $wk(P) = \max_L \left\lceil \frac{up(L)}{side(L)} \right\rceil$ where the maximum is taken over all forcing cycles L in P.*

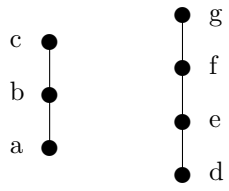


FIG. 2.2. *The ordered set* $\underline{3} + \underline{4}$.

Before proving Theorem 2.1, we return to the ordered set Q in Figure 2.1. It is easy to check that $wk(Q) = 2$. As noted before, the forcing cycle $L' : v_0, v_1, \ldots, v_6, v_0$ has $up(L') = 4$ and $side(L') = 3$, which yields $\left\lceil \frac{up(L')}{side(L')} \right\rceil = \lceil \frac{4}{3} \rceil = 2 = wk(Q)$.

Proof of Theorem 2.1. First we show $wk(P) \geq \left\lceil \frac{up(L)}{side(L)} \right\rceil$ for each forcing cycle L in P. Let $L : x = v_0, v_1, \ldots, v_m = x$ be a forcing cycle in P. To shorten notation, write $a = up(L)$, $b = side(L)$, $t = wk(P)$, and let lev be a t-leveling of P. Consider the sequence of differences: $S : lev(v_1) - lev(v_0), lev(v_2) - lev(v_1), \ldots, lev(v_m) - lev(v_{m-1})$. By the definition of a t-leveling function, if $v_i \prec v_{i+1}$, then $lev(v_{i+1}) - lev(v_i) \geq 1$, and if $v_i \parallel v_{i+1}$, then $lev(v_{i+1}) - lev(v_i) \geq -t$.

Clearly, the sum of the terms of S is 0; thus

$$0 = \sum_{i=0}^{m-1} [lev(v_{i+1}) - lev(v_i)] \geq a \cdot 1 + b \cdot (-t).$$

By our third remark listed above, we know b is at least 2; hence it is positive and thus we can conclude $t \geq a/b$. Since $t = wk(P)$ is an integer, we take the ceiling of the ratio to obtain $wk(P) = t \geq \lceil \frac{a}{b} \rceil = \left\lceil \frac{up(L)}{side(L)} \right\rceil$.

It remains to show that if $wk(P) = t$, then there exists a forcing cycle L in P so that $\left\lceil \frac{up(L)}{side(L)} \right\rceil = t$. We will construct such a cycle L using the algorithm k-Weak Leveling, which is given in [5]. We outline the steps of the algorithm that are needed here; a proof of the correctness of the algorithm and a time complexity bound can be found in [5].

Algorithm k-Weak Leveling begins by breaking an ordered set into *inseparable* suborders, where $Q = (X, \prec)$ is *inseparable* if it can *not* be partitioned nontrivially $X = V \cup W$ so that $v \prec w$ for every $v \in V$, $w \in W$. It is easy to show that the weakness of P is equal to the maximum of the weaknesses of these inseparable suborders. Thus without loss of generality, we will assume our ordered set P is inseparable. The algorithm constructs a k-leveling of P, if such a leveling exists, otherwise, it reports that P is not k-weak.

Since $wk(P) = t$, we know that P is *not* $(t-1)$-weak. We run Algorithm k-Weak Leveling on input P with $k = t - 1$.

Throughout the proof we will refer to the ordered set $H = \underline{3} + \underline{4}$ shown in Figure 2.2 and to Table 2.1 which shows an implementation of the algorithm k-Weak Leveling with $k = 2$ on the input H. It is easy to check that $wk(H) = 3$; thus the algorithm will fail to produce a 2-leveling of H. However, from this failed attempt, we will produce a forcing cycle L_1 in P with $\left\lceil \frac{up(L_1)}{side(L_1)} \right\rceil = 3$.

Given any k-leveling of an ordered set P, one can add the same constant to the level assigned to each element of P (i.e., $lev'(x) = lev(x) + c$ for each element x in P)

TABLE 2.1

A sequence of narrowing steps in the implementation of the Algorithm k-Weak Leveling with $k = 2$, applied to the ordered set in Figure 2.2.

z	Initial R(z)	NS 1 x=d y=e	NS 1 x=e y=f	NS 1 x=f y=g	NS 5 x=g y=a	NS 4 x=g y=a	NS 1 x=g y=f	NS 1 x=f y=e	NS 1 x=e y=d	NS 4 x=c y=d
a	$(-\infty,0]$				[0,0]					
b	[1,1]									
c	$[2, \infty)$									[2,1]
d	[-1,3]								[-1,-1]	
e	[-1,3]	[0,3]						[0,0]		
f	[-1,3]		[1,3]				[1,1]			
g	[-1,3]			[2,3]		[2,2]				

and the resulting function (lev') will be another k-leveling of P. Thus in searching for a k-leveling of P, we may start by picking a particular base element b of P and designating $lev(b) = 1$. This is the initial step in the algorithm.

Next, range sets $R(x) = [\ell(x), r(x)]$ are assigned to each $x \in X$ as follows. If $x = b$, then $R(x) = [1, 1]$. For $x \neq b$,

- if $b \prec x$, set $R(x) = [2, \infty)$.
- if $x \prec b$, set $R(x) = (-\infty, 0]$.
- if $b \parallel x$, set $R(x) = [1 - k, 1 + k]$.

By the definition of a k-leveling, if P has a k-leveling lev with $lev(b) = 1$, then for each element x in P, we have $lev(x) \in R(x)$. This statement will remain true even as the ranges are narrowed in the following steps. The second column of Table 2.1 shows the initial ranges for the elements of H.

After this initialization, the ranges are repeatedly narrowed by choosing pairs of distinct elements $x, y \in X - \{b\}$ (labeled so that $x \prec y$ or $x \parallel y$) and applying one or more of the following narrowing steps.

NS 1. If $x \prec y$ and $\ell(y) \leq \ell(x)$, increase $\ell(y)$ to $\ell(x) + 1$.

NS 2. If $x \prec y$ and $r(x) \geq r(y)$, decrease $r(x)$ to $r(y) - 1$.

NS 3. If $x \parallel y$ and $r(y) \geq r(x) + k + 1$, decrease $r(y)$ to $r(x) + k$.

NS 4. If $x \parallel y$ and $r(x) \geq r(y) + k + 1$, decrease $r(x)$ to $r(y) + k$.

NS 5. If $x \parallel y$ and $\ell(y) \leq \ell(x) - k - 1$, increase $\ell(y)$ to $\ell(x) - k$.

NS 6. If $x \parallel y$ and $\ell(x) \leq \ell(y) - k - 1$, increase $\ell(x)$ to $\ell(y) - k$.

Note that in applying NS 1 and NS 2 to ranges with infinite bounds, we think of "$-\infty + 1 = -\infty$" and "$\infty - 1 = \infty$." The pairs x, y are chosen with some care to ensure that the algorithm runs in polynomial time. We omit the details here because they are not necessary in our proof.

Either Algorithm k-Weak Leveling produces a k-leveling of P (by setting $lev(v) = \ell(v)$ for each v once the ranges stabilize and become finite) or, eventually, some element $v \in U$ has its range narrowed to the empty set, i.e., $R(v) = [\ell(v), r(v)] = \emptyset$ so $\ell(v) > r(v)$, where $\ell(v)$ and $r(v)$ are integers. Since P is not k-weak in our case, the latter occurs.

Table 2.1 shows the narrowing of ranges which occurs as a sequence of narrowing steps is applied to pairs of elements in H. In this example, the algorithm halts when $R(c)$ is narrowed to the empty set with $\ell(c) = 2$ and $r(c) = 1$.

In general, the left and right endpoints of $R(v)$ change as the algorithm runs, so we need notation to refer to the values of $\ell(v)$ and $r(v)$ at specific times in the algorithm's implementation. Let r_0 be the value of $r(v)$ when the algorithm halts

with $R(v) = [\ell(v), r(v)] = \emptyset$. We will backtrack through the implementation of the algorithm to produce a sequence of elements $v = v_0, v_1, \ldots, v_{m-1}, v_m = b$ so that applying the narrowing steps to the pairs $x = v_i$, $y = v_{i+1}$ for $i = m-1$ down to $i = 0$ yields r_0 as the right endpoint of $R(v)$.

Trace back through the implementation of Algorithm k-Weak Leveling to find the narrowing step at which the right endpoint of $R(v)$ was lowered to r_0. Note that right endpoints of range sets are lowered using only NS 2, 3, and 4 when they are compared to the right endpoints of range sets of other elements. Since NS 3 and 4 are the same step except that the roles of x and y are exchanged, without loss of generality, we will assume NS 4 was used. Let v_1 be the element for which the right endpoint of $R(v)$ was lowered to r_0 when the pair $x = v$, $y = v_1$ was considered, and let r_1 be the right endpoint of $R(v_1)$ at that time.

If $v_0 \prec v_1$ (NS 2), then $r_0 = r_1 - 1$. If $v_0 \parallel v_1$ (NS 4), then $r_0 = r_1 + k$.

Continue tracing back in this fashion. More precisely, given the sequence $v_i, v_{i-1}, \ldots, v_1, v_0 = v$ and the corresponding sequence of right endpoints $r_i, r_{i-1}, \ldots, r_1, r_0$, we trace back an additional step as follows. Let v_{i+1} be the element for which the right endpoint of $R(v_i)$ was lowered to r_i when the pair $x = v_i$, $y = v_{i+1}$ was considered. Let r_{i+1} be the right endpoint of $R(v_{i+1})$ at this time. If $v_i \prec v_{i+1}$ (NS 2), then $r_i = r_{i+1} - 1$. If $v_i \parallel v_{i+1}$ (NS 4), then $r_i = r_{i+1} + k$.

Eventually, we trace back to the beginning of Algorithm k-Weak Leveling, i.e., $v_m = b$ for some m and thus $r_m = lev(b) = 1$. This produces the sequence

$$S : v = v_0, v_1, v_2, \ldots, v_m = b$$

which for each i satisfies

- $v_i \prec v_{i+1}$, in which case $r_i = r_{i+1} - 1$, or
- $v_i \parallel v_{i+1}$, in which case $r_i = r_{i+1} + k$.

In our specific example H, the sequence produced is c, d, e, f, g, a, b and the values of r_i are $r_0 = 1, r_1 = -1, r_2 = 0, r_3 = 1, r_4 = 2, r_5 = 0, r_6 = 1$.

Let $up(S) = \#\{i : v_i \prec v_{i+1}\}$ and let $side(S) = \#\{i : v_i \parallel v_{i+1}\}$. Then $r_0 = r_m - up(S) + k \cdot side(S) = 1 - up(S) + k \cdot side(S)$.

Similarly, construct the sequence which leads to the left endpoint ℓ_0 of $R(v)$:

$$T : v = w_0, w_1, w_2, \ldots, w_r = b$$

which for each j satisfies

- $w_j \succ w_{j+1}$, in which case $\ell_j = \ell_{j+1} + 1$, or
- $w_j \parallel w_{j+1}$, in which case $\ell_j = \ell_{j+1} - k$.

Again, in our specific example, the sequence produced is c, b and the values of ℓ_j are $\ell_0 = 2, \ell_1 = 1$.

Let $down(T) = \#\{j : w_j \succ w_{j+1}\}$ and let $side(T) = \#\{j : w_j \parallel w_{j+1}\}$. Then $\ell_0 = \ell_r + down(T) - k \cdot side(T) = 1 + down(T) - k \cdot side(T)$.

Appending S to the reversal of T gives the forcing cycle

$$L : b = w_r, w_{r-1}, \ldots, w_1, w_0 = v = v_0, v_1, \ldots, v_m = b$$

with $up(L) = up(S) + down(T)$ and $side(L) = side(S) + side(T)$. The cycle produced in our example is $L_1 : b, c, d, e, f, g, a, b$, with $up(L_1) = 5$ and $side(L_1) = 2$, which yields $\left\lceil \frac{up(L_1)}{side(L_1)} \right\rceil = 3 = wk(H)$.

In general, since $r_0 < \ell_0$, we have

$$\begin{aligned} 1 - up(S) + k \cdot side(S) = r_0 &< \ell_0 = 1 + down(T) - k \cdot side(T) \\ k \cdot side(L) &< up(L) \\ k &< \frac{up(L)}{side(L)}. \end{aligned}$$

However, k is an integer, so $\left\lceil \frac{up(L)}{side(L)} \right\rceil \geq k + 1 = t = wk(P)$. By the proof of the first half of this theorem, $wk(P) \geq \left\lceil \frac{up(L)}{side(L)} \right\rceil$. Thus $wk(P) = \left\lceil \frac{up(L)}{side(L)} \right\rceil$ and L is our desired forcing cycle. □

3. Upper bounds on weakness. Recall that the height of an ordered set P, denoted $hgt(P)$, is the number of elements in a maximum chain of P. In the introduction we observed that any ordered set P with n elements has $wk(P) \leq n - 1$. In this section we improve this bound in two different ways by proving that

- $wk(P) \leq hgt(P)$ in Proposition 3.1; and
- $wk(P) \leq \left\lceil \frac{n-2}{2} \right\rceil$ in Theorem 3.4.

The first is a simple result; our proof of the second uses forcing cycles. The inequalities in each of these results is sharp, and in each case we give examples where equality holds.

PROPOSITION 3.1. *If $P = (X, \prec)$ is an ordered set, then $wk(P) \leq hgt(P) - 1$.*

Proof. Let $P = P_1$ be an ordered set with $hgt(P) = m$. For $i = 1, 2, \ldots, m$, let S_i be the set of all minimal elements of P_i, and let $P_{i+1} = P_i - S_i$. Thus the sets $S_1, S_2, \ldots, S_m$ partition X. For $v \in S_i$, define $lev(v) = i$. It is easy to check that this is a valid $(m-1)$-leveling of P. □

The ordered set Q in Figure 2.1 has $hgt(Q) = 3$ and $wk(Q) = 2$; thus the inequality in Proposition 3.1 is sharp.

The next lemma is an easy algebraic result which we need in the proof of Lemma 3.3.

LEMMA 3.2. *Let a, b, c and d be nonnegative integers with b and d strictly greater than 0. If $a/b \leq c/d$, then $a/b \leq (a+c)/(b+d) \leq c/d$.*

Proof. The proof (by contradiction) is a simple calculation. □

Our definition of forcing cycles allows for a sequence with repeated elements. However, as the next lemma shows, forcing cycles with repeated elements are never needed. Thus we are motivated to define the following: A forcing cycle $L : x = v_0, v_1, v_2, \ldots, v_m = x$ is said to have *distinct elements* if the v_i are distinct (except for $v_0 = v_m$).

LEMMA 3.3. *If L is a forcing cycle of an ordered set P, then there exists a forcing cycle L' of P with distinct elements so that $\left\lceil \frac{up(L')}{side(L')} \right\rceil \geq \left\lceil \frac{up(L)}{side(L)} \right\rceil$.*

Proof. For a contradiction, assume that L is a smallest forcing cycle that fails to satisfy the lemma. If L had distinct elements, then $L' = L$ satisfies the lemma. Thus we may assume L has repeated elements.

Choose the starting point x of $L : x = v_0, v_1, \ldots, v_m = x$ so that there exists $i : 1 < i < m$ for which the elements $v_0, v_1, \ldots, v_{i-1}$ are all distinct, but that $v_i = v_0$. Then L can be broken into two smaller, nontrivial forcing cycles $L_1 : x = v_0, v_1, \ldots, v_i = x$, and $L_2 : v_{i+1}, v_{i+2}, \ldots, v_m = v_0 = v_i, v_{i+1}$. By construction, L_1 has distinct elements. We note that $up(L) = up(L_1) + up(L_2)$ and $side(L) = side(L_1) + side(L_2)$.

If $\frac{up(L_1)}{side(L_1)} \geq \frac{up(L_2)}{side(L_2)}$, then by Lemma 3.2 (with $a = up(L_2)$, $b = side(L_2)$, $c = up(L_1)$ and $d = side(L_1)$) we have $\frac{up(L)}{side(L)} \leq \frac{up(L_1)}{side(L_1)}$ and $L' = L_1$ satisfies the conditions of the lemma.

Otherwise, $\frac{up(L_1)}{side(L_1)} < \frac{up(L_2)}{side(L_2)}$. We again apply Lemma 3.2 (this time with $a = up(L_1)$, $b = side(L_1)$, $c = up(L_2)$, and $d = side(L_2)$) and conclude $\frac{up(L)}{side(L)} \leq \frac{up(L_2)}{side(L_2)}$. By construction, L_2 is smaller than L, so there exists a forcing cycle L_3 with distinct elements in P such that $\left\lceil \frac{up(L_3)}{side(L_3)} \right\rceil \geq \left\lceil \frac{up(L_2)}{side(L_2)} \right\rceil$. In this case, choose $L' = L_3$.

Now in either case, L' is a forcing cycle with distinct elements in P for which $\left\lceil \frac{up(L')}{side(L')} \right\rceil \geq \left\lceil \frac{up(L)}{side(L)} \right\rceil$ as desired. □

THEOREM 3.4. *If P is an ordered set with n elements, then $wk(P) \leq \lceil \frac{n-2}{2} \rceil$.*

Proof. Let P be an ordered set with n elements and let $k = wk(P)$. By Theorem 2.1, there exists a forcing cycle L in P with $\left\lceil \frac{up(L)}{side(L)} \right\rceil = k$. By Lemma 3.3, there exists a forcing cycle L' of P with distinct elements so that $\left\lceil \frac{up(L')}{side(L')} \right\rceil \geq k$. Let L' be the sequence $L' : x = v_0, v_1, \ldots, v_m = x$.

Since k is an integer, $\frac{up(L')}{side(L')} > k - 1$; thus $up(L') > (side(L'))(k-1) \geq 2k - 2$, because $side(L) \geq 2$ for any forcing cycle L. Since $up(L')$ and $2k-2$ are integers, $up(L') \geq 2k - 1$. Thus $m = up(L') + side(L') \geq (2k-1) + 2 = 2k + 1$.

Recall that L' has distinct elements, so P has at least $2k + 1$ elements, i.e., $n \geq 2k + 1$. If n is odd, $k \leq \frac{n-1}{2} = \lceil \frac{n-2}{2} \rceil$. If n is even, $n \geq 2k + 1 \Longrightarrow n \geq 2k + 2$; thus $k \leq \frac{n-2}{2} = \lceil \frac{n-2}{2} \rceil$. So in either case, $wk(P) = k \leq \lceil \frac{n-2}{2} \rceil$. □

The following example shows that the inequality in Theorem 3.4 is sharp. Let $r = \lceil \frac{n}{2} \rceil$ and $s = \lfloor \frac{n}{2} \rfloor$. Then, the ordered set $P = \underline{r} + \underline{s}$ has exactly n elements. All forcing cycles L in P with distinct elements have $side(L) \geq 2$. The cycle L with the largest value of $up(L)$ uses each element of P exactly once. Therefore, $up(L) = (\lceil \frac{n}{2} \rceil - 1) + (\lfloor \frac{n}{2} \rfloor - 1) = n - 2$. Using Theorem 2.1, we have $wk(P) = \left\lceil \frac{up(L)}{side(L)} \right\rceil = \lceil \frac{n-2}{2} \rceil$.

4. Computing the weakness of an ordered set. In [5] it is shown that Algorithm k-Weak Leveling determines if an n-element order P is k-weak in $O(n^4 k)$ time. Combining Algorithm k-Weak Leveling with Theorem 3.4 immediately yields the following $O(n^6)$ algorithm for computing the weakness of an n-element ordered set.

Use Algorithm k-Weak Leveling to check if P is i-weak for $i = 0, 1, 2, \ldots$ and stop as soon as a value of i is found for which P is i-weak. By definition, this value of i is $wk(P)$. Theorem 3.4 ensures that in the worst case, Algorithm k-Weak Leveling is implemented for $i = 1, 2, \ldots, \lceil \frac{n-2}{2} \rceil$ before finding an i for which P is i-weak. Thus in the worst case we run an $O(n^4 k)$ algorithm for $k = 1, 2, \ldots, \lceil \frac{n-2}{2} \rceil$, resulting in a complexity of $O(n^6)$.

A better method was suggested by Jeremy Spinrad (personal communication) and implemented by Christina Chen [2]. This method searches for a forcing cycle L that maximizes the ratio $\left\lceil \frac{up(L)}{side(L)} \right\rceil$. For each value of $r = 2, 3, \ldots, n$ the algorithm uses dynamic programming to compute the maximum value of $up(L)$ when $side(L)$ is fixed at r. The running time for this algorithm is $O(n^4)$. Once the value of $k = wk(P)$ is computed, Algorithm k-Weak Leveling is run to construct a k-leveling of P.

5. Weakness is a comparability invariant. If $P = (V, \prec)$ is an ordered set, its associated comparability graph $G = (V, E)$ is the graph whose vertex set is the

element set of P and with $xy \in E$ iff $x \prec y$ or $y \prec x$. In this section we follow the notation of [6].

Given a graph $G = (V, E)$, an *autonomous set* $S \subseteq V$ is one with the property that every vertex $x \in V - S$ is either adjacent to all vertices in S or to none of the vertices in S. Autonomous sets play an important role in relating two ordered sets that have the same comparability graph.

Let $G = (V, E)$ be a comparability graph and let $P_1 = (V, \prec_1)$ and $P_2 = (V, \prec_2)$ be ordered sets, each of whose associated comparability graph is G. We say that P_2 is obtained from P_1 by an *elementary reversal* if there exists an autonomous set S of G such that

- S is not an independent set;
- if x, y are not both in S, then $x \prec_1 y$ iff $x \prec_2 y$; and
- if $x, y \in S$, then $x \prec_1 y$ iff $y \prec_2 x$.

Thus to obtain P_2 from P_1 (or vice versa), one reverses all comparabilities within S and leaves all other comparabilities/incomparabilities unchanged. The first of the three conditions ensures that $P_1 \neq P_2$.

Several parameters of ordered sets are known to be comparability invariant, most notably, dimension [6]. The next theorem shows that weakness is a comparability invariant.

THEOREM 5.1. *Let $G = (V, E)$ be the comparability graph associated with ordered sets P and Q. Then $wk(P) = wk(Q)$.*

We will need the following result of Gallai [3] (which appears in [6, pgs. 61–62]) for the proof of Theorem 5.1. A simple proof of Lemma 5.2 appears in [4].

LEMMA 5.2. *Let $G = (V, E)$ be the comparability graph associated with distinct ordered sets $P = (V, \prec_p)$ and $Q = (V, \prec_q)$. Then, there exists a sequence of ordered sets $P_0, P_1, \ldots, P_m$ so that $P_0 = P$, $P_m = Q$ and P_{i+1} is obtained from P_i by an elementary reversal for $i = 0, 1, \ldots, m-1$.*

Proof of Theorem 5.1. Let $G = (V, E)$ be a comparability graph and let P and Q be ordered sets, each of whose associated comparability graph is G. By Lemma 5.2, there exist orders $P_0 = P, P_1, \ldots, P_m = Q$ on set V such that P_{i+1} is obtained from P_i by an elementary reversal for $i = 0, 1, \ldots, m-1$. Thus it suffices to show $wk(P_i) = wk(P_{i+1})$. We accomplish this by showing that if P_i has a k-leveling function, then so does P_{i+1}. This suffices to prove the theorem, since the process of obtaining one ordered set from another by an elementary reversal is a symmetric operation.

Let $P_i = (V, \prec_1)$ and $P_{i+1} = (V, \prec_2)$. Since P_{i+1} is obtained from P_i by an elementary reversal, there exists an autonomous set $S \subseteq V$ of G associated with this reversal. Let $lev_1 : V \to \mathbf{Z}$ be a k-leveling function of P_i. Let $s = \min\{lev_1(v) : v \in S\}$ and let $t = \max\{lev_1(v) : v \in S\}$. We construct the function $lev_2 : V \to \mathbf{Z}$ as follows:

- If $x \in S$, let $lev_2(x) = s + t - lev_1(x)$.
- If $x \in V - S$, let $lev_2(x) = lev_1(x)$.

To complete the proof, one checks that lev_2 is a valid k-leveling function of P_{i+1}. □

Acknowledgments. The authors would like to thank Ed Scheinerman and Jeremy Spinrad for helpful conversations on this subject.

REFERENCES

[1] K. P. Bogart, *Introductory Combinatorics*, Harcourt Brace Jovanovich, New York, 1980.
[2] C. S. Chen, *An Improved Algorithm for Computing Weakness*, preliminary report, Wellesley College, Wellesley, MA, 1997.
[3] T. Gallai, *Transitiv orientierbare Graphen*, Acta Math. Hungar., 18 (1967), pp. 25–66.
[4] J. Gimbel and C. Thomassen, *Partial orders on a fixed graph*, in Contemporary Methods in Graph Theory, R. Bodendiek, ed., Wissenschaftsverlag, Mannheim, 1990, pp. 305–312.
[5] A. N. Trenk, *On k-weak orders: Recognition and a tolerance result*, Discrete Math., 181 (1998), pp. 223–237.
[6] W. T. Trotter, *Combinatorics and Partially Ordered Sets*, Johns Hopkins University Press, Baltimore, MD, 1992.

SIAM J. DISCRETE MATH.
Vol. 11, No. 4, pp. 664–671, November 1998

011

A NEW DECODING ALGORITHM FOR COMPLETE DECODING OF LINEAR BLOCK CODES*

YUNGHSIANG S. HAN†

Abstract. In this paper we present and describe an improved version of the Zero-Neighbors algorithm, which we call the Zero-Coverings algorithm. We also present a method for finding a smallest subset of codewords (Zero-Coverings) which need to be stored to perform the Zero-Coverings algorithm. For some short codes, the sizes of Zero-Coverings are obtained by computer searches; for long codes, an asymptotic bound on the sizes of such subsets is also given.

Key words. coding, decoding, linear codes, block codes

AMS subject classifications. 94B35, 94B05

PII. S0895480197323974

1. Introduction. In general, complete decoding [11] for a linear block code has proved to be an NP-hard computational problem [1]. That is, it is unlikely that a polynomial time (space) complete decoding algorithm for a linear block code can be found. A new decoding algorithm, the Zero-Neighbors algorithm (ZNA) [9], using the concept of a Zero-Neighbors, was proposed. Only the codewords in a Zero-Neighbors need to be stored and used in the decoding procedure. The size of a Zero-Neighbors is very small compared to $\min(2^k, 2^{n-k})$ for $n >> 1$ and a wide range of code rates $R = k/n$. An improvement of the Zero-Neighbors algorithm, the Zero-Guards algorithm (ZGA), was recently presented [7, 10]. The ZGA further reduces the number of codewords to be stored. The special set of these codewords is called *Zero-Guards*. The time and space complexity of the ZNA and ZGA are determined by the sizes of the Zero-Neighbors and the Zero-Guards used, respectively. The problem here is how to find the smallest subset of codewords that can be used to perform the ZNA-like decoding procedure. We call all the decoding algorithms that perform a ZNA-like decoding procedure "ZNA-like" algorithms. Similarly, we call any subset of codewords that can be used to perform a ZNA-like algorithm procedure a "ZN-like" subset of codewords. The ZN-like subset of codewords with the smallest size is called an "optimal ZN-like set." Furthermore, a ZNA-like algorithm using an optimal ZN-like set is denoted as an "optimal ZNA-like" algorithm.

In this paper we present an optimal ZNA-like algorithm, the Zero-Coverings algorithm, and give a systematic way in which to find an optimal ZN-like set, a Zero-Coverings. Furthermore, an asymptotic bound on the size of an optimal ZN-like set is derived for long codes. In section 2 we briefly review the Zero-Neighbors and the Zero-Guards algorithms. In section 3 we give a description of the Zero-Coverings algorithm and, in the next section, properties of Zero-Coverings are presented. We also give a systematic way to find Zero-Coverings. Simulation results and an asymptotic bound on the size of a Zero-Coverings are given in section 5. Remarks and conclusions are given in section 6.

*Received by the editors July 7, 1997; accepted for publication (in revised form) January 29, 1998; published electronically September 1, 1998. This work was supported by National Science Council ROC grant NSC 87-2218-E-260-002. Portions of this research were presented at the IEEE International Symposium on Information Theory, Ulm, Germany, June 1997.

http://www.siam.org/journals/sidma/11-4/32397.html

†Department of Computer Science and Information Engineering, National Chi Nan University, Puli NanTou, Taiwan, 545 R.O.C. (yshan@csie.ncnu.edu.tw).

2. The Zero-Neighbors and the Zero-Guards algorithms. In this section we briefly describe the ZNA and an improved version of it, the ZGA. First, we give some definitions.

Let $\boldsymbol{Z}$ be the set of all binary vectors of length n, and let $\boldsymbol{C} \subset \boldsymbol{Z}$ be a binary linear block code. Let $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ denote the Hamming distance between $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{Z}$. Let $w(\boldsymbol{x}) = d(\boldsymbol{x}, \boldsymbol{0})$ denote the Hamming weight of $\boldsymbol{x}$ and let $\oplus$ denote the modulo-2 addition. Furthermore, let $d_{\min}$ be the nonzero minimum weight of codewords in $\boldsymbol{C}$. In this paper we will assume that $d_{\min} \geq 2$.

DEFINITION 2.1. *The domain $D(\boldsymbol{c})$ of a codeword $\boldsymbol{c} \in \boldsymbol{C}$ is the set of all $\boldsymbol{x} \in \boldsymbol{Z}$ such that $d(\boldsymbol{x}, \boldsymbol{c}) \leq d(\boldsymbol{x}, \boldsymbol{c}')$, for all $\boldsymbol{c}' \in \boldsymbol{C}$.*

DEFINITION 2.2. *The vicinity $B(\boldsymbol{x})$ of $\boldsymbol{x} \in Z$ is the set of all $\boldsymbol{y} \in Z$ such that $d(\boldsymbol{x}, \boldsymbol{y}) = 1$. The domain frame $G(\boldsymbol{c})$ of a codeword $\boldsymbol{c} \in \boldsymbol{C}$ is the set $G(\boldsymbol{c}) = \bigcup_{\boldsymbol{x} \in D(\boldsymbol{c})} B(\boldsymbol{x}) - D(\boldsymbol{c})$.*

DEFINITION 2.3. *A set of Zero-Neighbors (ZN) is a set N_0 of codewords such that*

$$G(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N_0} D(\boldsymbol{c}), \text{ where}$$

$$|N_0| = \min \left\{ |N| \mid N \subset \boldsymbol{C}, G(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N} D(\boldsymbol{c}) \right\}.$$

It can be shown that if $\boldsymbol{x} \notin D(\boldsymbol{0})$, there exists a $\boldsymbol{c} \in N_0$ such that $w(\boldsymbol{x} \oplus \boldsymbol{c}) < w(\boldsymbol{x})$. Thus, the Zero-Neighbors algorithm is as follows.

Algorithm. Let $\boldsymbol{y} = \boldsymbol{y}_0 \in \boldsymbol{Z}$ be the received vector to be decoded. At the ith step of the algorithm we calculate $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c})$ for all $\boldsymbol{c} \in N_0$. If there exists a $\boldsymbol{c}_i \in N_0$ such that $w(\boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i) < w(\boldsymbol{y}_{i-1})$, we set $\boldsymbol{y}_i = \boldsymbol{y}_{i-1} \oplus \boldsymbol{c}_i$ and go to the next step; otherwise, the algorithm terminates. If the algorithm terminates at the $(m+1)$th step, then $\boldsymbol{y}_m = \boldsymbol{y} \oplus \sum_{i=1}^{m} \boldsymbol{c}_i \in D(\boldsymbol{0})$ and can be taken as a coset leader, while $\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{c}_i \in \boldsymbol{C}$ is a codeword that is one of the closest to $\boldsymbol{y}$.

We need only to store a ZN to accomplish this algorithm. It can be shown that the number of steps m mentioned above is less than or equal to $n - \lfloor \frac{d_{\min}}{2} \rfloor$. Furthermore, if $\boldsymbol{1}$ is in $\boldsymbol{C}$, then $m \leq \lfloor \frac{n+1}{2} \rfloor$. Another improved version of the ZNA, the ZGA, is described next.

DEFINITION 2.4. *The frontier $F(\boldsymbol{0})$ of $\boldsymbol{0}$ is the set of all $\boldsymbol{x} \in \boldsymbol{Z}$ such that all its proper descendants* [12] *belong to $D(\boldsymbol{0})$ and $\boldsymbol{x} \notin D(\boldsymbol{0})$.*

DEFINITION 2.5. *A Zero-Guards (ZG) is a set RN_0 of codewords such that*

$$F(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in RN_0} D(\boldsymbol{c}), \text{ where}$$

$$|RN_0| = \min \left\{ |N| \mid N \subset \boldsymbol{C}, F(\boldsymbol{0}) \subset \bigcup_{\boldsymbol{c} \in N} D(\boldsymbol{c}) \right\}.$$

In other words, the set of domains of codewords in RN_0 forms a minimum covering of $F(\boldsymbol{0})$. It is not difficult to see that $F(\boldsymbol{0}) \subset G(\boldsymbol{0})$. Consequently, the number of codewords in a ZG is less than or equal to that in a ZN. The decoding procedure of the Zero-Neighbors algorithm described above can be applied to the Zero-Guards algorithm while we use a ZG instead of a ZN in the procedure.

3. An optimal ZN-like set. In this section we will give a systematic way to find an optimal ZN-like set, a Zero-Coverings (ZC), which is related to a Zero-Guards. First, we give a formal definition of a ZN-like subset of codewords.

DEFINITION 3.1. *A ZN-like subset of codewords, C_{ZN}, is a subset of $\boldsymbol{C}$ with the following property: for every received vector $\boldsymbol{y}$, if $\boldsymbol{y} \notin D(\mathbf{0})$, then there exists a $\boldsymbol{c} \in C_{ZN}$ such that $w(\boldsymbol{y} \oplus \boldsymbol{c}) < w(\boldsymbol{y})$.*

It has been shown that a ZN and a ZG are ZN-like subsets of codewords in [9] and [6], respectively. It is not difficult to see that if N_0 in the algorithm given in section 2 is replaced with C_{ZN}, the algorithm will still perform complete decoding. That is, the algorithm is a ZNA-like algorithm. Since the time and space complexity of any ZNA-like algorithm grow with the size of C_{ZN}, in order to reduce the complexity we need to find the smallest C_{ZN}.

DEFINITION 3.2. *The covering domain $D_c(\boldsymbol{c})$ of a codeword $\boldsymbol{c} \in \boldsymbol{C}$ is the set of all $\boldsymbol{x} \in F(\mathbf{0})$ such that $d(\boldsymbol{x}, \boldsymbol{c}) < d(\boldsymbol{x}, \mathbf{0})$.*

That is, $D_c(\boldsymbol{c})$ contains all vectors in the frontier $F(\mathbf{0})$ such that they are closer to $\boldsymbol{c}$ than to $\mathbf{0}$. Furthermore, if $\boldsymbol{x} \in D(\boldsymbol{c})$, then $\boldsymbol{x} \in D_c(\boldsymbol{c})$ for any $\boldsymbol{x} \in F(\mathbf{0})$.

DEFINITION 3.3. *A set of Zero-Coverings (ZC) is a subset of $\boldsymbol{C}$ such that*

$$F(\mathbf{0}) = \bigcup_{\boldsymbol{c} \in ZC} D_c(\boldsymbol{c}), \text{ where} \tag{1}$$

$$|ZC| = \min \left\{ |N| \mid N \subset \boldsymbol{C}, F(\mathbf{0}) = \bigcup_{\boldsymbol{c} \in N} D_c(\boldsymbol{c}) \right\}. \tag{2}$$

In other words, the set of covering domains of a Zero-Coverings forms a minimum covering of the frontier $F(\mathbf{0})$. The algorithm for solving general minimum covering problems can be found in [5].

There are many properties of the frontier $F(\mathbf{0})$, derived in [6], that can help us to find $F(\mathbf{0})$. We state these properties here without proofs. The details of these properties can be found in [6].

LEMMA 3.4. *Let $S(\boldsymbol{x}, a) = \{\boldsymbol{v} \mid \boldsymbol{v} \in \boldsymbol{Z}, w(\boldsymbol{v}) = a$ and $\boldsymbol{v}$ be a descendant of $\boldsymbol{x}\}$. Then $\boldsymbol{x} \in F(\mathbf{0})$ iff $\boldsymbol{x} \notin D(\mathbf{0})$ and $S(\boldsymbol{x}, w(\boldsymbol{x}) - 1) \subset D(\mathbf{0})$.*

LEMMA 3.5. *If $\boldsymbol{x} \in F(\mathbf{0})$, then there exists at least one $\boldsymbol{c} \in \boldsymbol{C}$ such that $\boldsymbol{x} \in D(\boldsymbol{c})$ and $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.*

LEMMA 3.6. *Let $\boldsymbol{x} \in F(\mathbf{0})$. If $d(\boldsymbol{x}, \boldsymbol{c}) < w(\boldsymbol{x})$, then $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.*

LEMMA 3.7. *Let $\boldsymbol{y} \in \boldsymbol{Z}$ and $\boldsymbol{y} \notin D(\mathbf{0})$. Then there exists a descendant $\boldsymbol{x}$ of $\boldsymbol{y}$ such that $\boldsymbol{x} \in F(\mathbf{0})$.*

LEMMA 3.8. *For every $\boldsymbol{c} \in \boldsymbol{C}$ and $\boldsymbol{c} \neq \mathbf{0}$ there exists a descendant $\boldsymbol{x}$ of $\boldsymbol{c}$ such that $\boldsymbol{x} \in F(\mathbf{0})$.*

The following are some new results that are related to covering domains.

LEMMA 3.9. *If $\boldsymbol{x} \in F(\mathbf{0})$, then there exists at least one $\boldsymbol{c} \in \boldsymbol{C}$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c})$ and $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.*

Proof. Since $\boldsymbol{x} \in F(\mathbf{0})$ and $\boldsymbol{x} \in D(\boldsymbol{c})$ imply that $\boldsymbol{x} \in D_c(\boldsymbol{c})$, by Lemma 3.5, the result holds. □

LEMMA 3.10. *If $\boldsymbol{x} \in D_c(\boldsymbol{c})$, then $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$.*

Proof. The result follows directly from Lemma 3.6. □

LEMMA 3.11. *If $\boldsymbol{c} \in ZC$, then there exists one $\boldsymbol{x} \in F(\mathbf{0})$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c})$ and $\boldsymbol{x} \notin D_c(\boldsymbol{c}')$, $\boldsymbol{c}' \neq \boldsymbol{c}$, $\boldsymbol{c}' \in ZC$.*

Proof. Assume that there is no $\boldsymbol{x} \in F(\mathbf{0})$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c})$ and $\boldsymbol{x} \notin D_c(\boldsymbol{c}'), \boldsymbol{c}' \neq \boldsymbol{c}$, $\boldsymbol{c}' \in ZC$. Then for every $\boldsymbol{x} \in D_c(\boldsymbol{c})$ and $\boldsymbol{x} \in F(\mathbf{0})$, there exists at least one $\boldsymbol{c}' \in ZC$, $\boldsymbol{c}' \neq \boldsymbol{c}$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c}')$. Therefore, if we remove $\boldsymbol{c}$ from ZC we also have $F(\mathbf{0}) = \bigcup_{\boldsymbol{c} \in ZC} D_c(\boldsymbol{c})$. The above result contradicts the statement that ZC is a minimum set such that $F(\mathbf{0}) = \bigcup_{\boldsymbol{c} \in ZC} D_c(\boldsymbol{c})$. □

Next we need to prove that a ZC is a ZN-like subset of codewords. In order to show this, it is sufficient to prove the following theorem.

THEOREM 3.12. *$\boldsymbol{y} \notin D(\mathbf{0})$ iff there exists one $\boldsymbol{c} \in ZC$ such that $w(\boldsymbol{y} \oplus \boldsymbol{c}) < w(\boldsymbol{y})$.*

Proof. Assume that $\boldsymbol{y} \notin D(\mathbf{0})$. From Lemma 3.7, there exists a descendant $\boldsymbol{x}$ of $\boldsymbol{y}$ such that $\boldsymbol{x} \in F(\mathbf{0})$. Consider a $\boldsymbol{c} \in ZC$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c})$. Hence, $w(\boldsymbol{y} \oplus \boldsymbol{c}) = d(\boldsymbol{y}, \boldsymbol{c}) \leq d(\boldsymbol{y}, \boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{c}) < d(\boldsymbol{y}, \boldsymbol{x}) + d(\boldsymbol{x}, \mathbf{0}) = w(\boldsymbol{y})$. Assume that $\boldsymbol{y} \in D(\mathbf{0})$. Then $d(\boldsymbol{y}, \mathbf{0}) \leq d(\boldsymbol{y}, \boldsymbol{c})$ for all $\boldsymbol{c} \in \boldsymbol{C}$. Thus, $w(\boldsymbol{y}) \leq w(\boldsymbol{y} \oplus \boldsymbol{c})$ and no $\boldsymbol{c} \in ZC$, such that $d(\boldsymbol{y} \oplus \boldsymbol{c}) < w(\boldsymbol{y})$. □

Now we prove that ZC is an optimal ZN-like set.

THEOREM 3.13. *A Zero-Coverings is an optimal ZN-like set.*

Proof. Assume that we have a ZN-like subset of codewords, C_{ZN}. Let $\boldsymbol{x} \in F(\mathbf{0})$. Since $\boldsymbol{x} \notin D(\mathbf{0})$, by the properties of C_{ZN}, there exists one $\boldsymbol{c} \in C_{ZN}$ such that $d(\boldsymbol{x}, \boldsymbol{c}) < d(\boldsymbol{x}, \mathbf{0})$. Therefore, $\boldsymbol{x} \in D_c(\boldsymbol{c})$. If we run through all of the elements in $F(\mathbf{0})$, we have a subset of C_{ZN}, denoted as C'_{ZN}, such that

$$F(\mathbf{0}) = \bigcup_{\boldsymbol{c} \in C'_{ZN}} D_c(\boldsymbol{c}).$$

Consequently, any ZN-like subset of codewords will contain a subset that satisfies the above equality. Therefore, by Definition 3.3, a ZC is a ZN-like subset of codewords with the smallest size that satisfies the above equality. □

In general, the ZGA is not an optimal ZNA-like algorithm. One example to illustrate this fact is given in the appendix.

4. Properties of the frontier of 0 and a Zero-Coverings. In this section we give some theorems describing the properties of the frontier of $\mathbf{0}$ and a ZC that can be used to find the ZC.

DEFINITION 4.1. *Let $\boldsymbol{xC}$ be the coset containing $\boldsymbol{x}$. Furthermore, let $w(\boldsymbol{xC})$ be the Hamming weight of a coset leader in $\boldsymbol{xC}$.*

THEOREM 4.2. *$\boldsymbol{x} \in F(\mathbf{0})$ iff $w(\boldsymbol{x}) - 2 \leq w(\boldsymbol{xC}) \leq w(\boldsymbol{x}) - 1$ and for every vector $\boldsymbol{v}$ in $\boldsymbol{xC}$ with $w(\boldsymbol{v}) < w(\boldsymbol{x})$, $w(\boldsymbol{x} \oplus \boldsymbol{v}) = w(\boldsymbol{x}) + w(\boldsymbol{v})$.*

Proof. Assume that $\boldsymbol{x} \in F(\mathbf{0})$. Since $w(\boldsymbol{xC}) < w(\boldsymbol{x})$, then $w(\boldsymbol{xC}) \leq w(\boldsymbol{x}) - 1$. Furthermore, assume that $w(\boldsymbol{xC}) < w(\boldsymbol{x}) - 2$. Let $\boldsymbol{u}$ be a coset leader in $\boldsymbol{xC}$ and Let $\boldsymbol{v}_1$ be an immediate descendant of $\boldsymbol{x}$ which differs from $\boldsymbol{x}$ in the ith position. Furthermore, let $\boldsymbol{v}_2$ be a vector that differs from $\boldsymbol{u}$ only in the ith position. Then $w(\boldsymbol{v}_2) \leq w(\boldsymbol{x}) - 2$ and $w(\boldsymbol{v}_1) = w(\boldsymbol{x}) - 1$. Since $\boldsymbol{u}$ and $\boldsymbol{x}$ are in the same coset, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are also in the same coset. Thus, $\boldsymbol{v}_1 \notin D(\mathbf{0})$. This contradicts the statement that $\boldsymbol{v}_1 \in D(\mathbf{0})$.

Assume that $w(\boldsymbol{x} \oplus \boldsymbol{v}) \neq w(\boldsymbol{x}) + w(\boldsymbol{v})$ for a vector $\boldsymbol{v}$ in $\boldsymbol{xC}$, where $w(\boldsymbol{v}) < w(\boldsymbol{x})$. Then there are two cases to consider:

1. $w(\boldsymbol{v}) = w(\boldsymbol{xC})$. Since $w(\boldsymbol{x} \oplus \boldsymbol{v}) \neq w(\boldsymbol{x}) + w(\boldsymbol{v})$, there exists a position such that $\boldsymbol{x}$ and $\boldsymbol{v}$ are one in that position. Let $\boldsymbol{v}_3$ and $\boldsymbol{v}_4$ be descendants of $\boldsymbol{x}$ and $\boldsymbol{v}$, which differ from them in the position just mentioned, respectively. Since $\boldsymbol{x}$ and $\boldsymbol{v}$ are in the same coset, then $\boldsymbol{v}_3$ and $\boldsymbol{v}_4$ are in the same coset, also. Obviously, $w(\boldsymbol{v}_3) > w(\boldsymbol{v}_4)$. This contradicts the statement that $\boldsymbol{v}_3 \in D(\mathbf{0})$.

2. $w(\boldsymbol{v}) \neq w(\boldsymbol{xC})$. Then $w(\boldsymbol{v}) = w(\boldsymbol{x}) - 1$ and $w(\boldsymbol{xC}) = w(\boldsymbol{x}) - 2$. In this case, the argument is similar to that above.

Now assume that, for every vector $\boldsymbol{v}$ in $\boldsymbol{xC}$ with $w(\boldsymbol{v}) < w(\boldsymbol{x})$, $w(\boldsymbol{x} \oplus \boldsymbol{v}) = w(\boldsymbol{x}) + w(\boldsymbol{v})$ and $w(\boldsymbol{x}) - 2 \leq w(\boldsymbol{xC}) \leq w(\boldsymbol{x}) - 1$. We want to prove that $\boldsymbol{x} \in F(\boldsymbol{0})$. That is, we need to prove that every immediate descendant of $\boldsymbol{x}$ belongs to $D(\boldsymbol{0})$. Let $\boldsymbol{v}$ be any vector in $\boldsymbol{xC}$ such that $w(\boldsymbol{v}) < w(\boldsymbol{x})$. Let $\boldsymbol{v}_5$ be an immediate descendant of $\boldsymbol{x}$ that differs from $\boldsymbol{x}$ in the ith position. Let $\boldsymbol{v}_6$ be a vector that is one in the ith position and that differs from $\boldsymbol{v}$ only in that position. Therefore, $\boldsymbol{v}_5$ and $\boldsymbol{v}_6$ are in the same coset. $\boldsymbol{v}_6$ has a weight of at least $w(\boldsymbol{x}) - 1$ since $w(\boldsymbol{x} \oplus \boldsymbol{v}) = w(\boldsymbol{x}) + w(\boldsymbol{v})$. Therefore, $\boldsymbol{v}_5 \in D(\boldsymbol{0})$. □

Base on the above theorem, we can design an efficient algorithm to find $F(\boldsymbol{0})$ from a standard array. Furthermore, we can find the $D_c(\boldsymbol{c})$ from a standard array by the following theorems. Since the proofs of the theorems are simple, we omit them here.

THEOREM 4.3. *Let $\boldsymbol{x} \in F(\boldsymbol{0})$; $\boldsymbol{x} \in D_c(\boldsymbol{c})$ iff there exists a vector $\boldsymbol{v}$ in $\boldsymbol{xC}$ such that $w(\boldsymbol{v}) < w(\boldsymbol{x})$ and $\boldsymbol{c} = \boldsymbol{v} \oplus \boldsymbol{x}$. Furthermore, if $\boldsymbol{x} \in D_c(\boldsymbol{c})$, then $w(\boldsymbol{c}) = 2w(\boldsymbol{x}) - 2$ or $w(\boldsymbol{c}) = 2w(\boldsymbol{x}) - 1$.*

THEOREM 4.4. *Let $\boldsymbol{x} \in F(\boldsymbol{0})$ and $\boldsymbol{x} \in D_c(\boldsymbol{c})$; then $w(\boldsymbol{xC}) \leq d(\boldsymbol{x}, \boldsymbol{c}) \leq w(\boldsymbol{xC})+1$.*

The following result can be used to derive an upper bound on the size of a ZC.

THEOREM 4.5. *Let r be the covering radius of the code $\boldsymbol{C}$. If $\boldsymbol{c} \in \boldsymbol{C}$ and $w(\boldsymbol{c}) > 2r + 1$, then $\boldsymbol{c} \notin ZC$.*

Proof. Assume that $\boldsymbol{c} \in ZC$. From Lemma 3.11 there exists an $\boldsymbol{x} \in F(\boldsymbol{0})$, $\boldsymbol{x} \in D_c(\boldsymbol{c})$, and $\boldsymbol{x} \notin D_c(\boldsymbol{c}'), \boldsymbol{c}' \neq \boldsymbol{c}$. Since $\boldsymbol{x} \in F(\boldsymbol{0}), w(\boldsymbol{x}) \leq r + 1$ and $d(\boldsymbol{x}, \boldsymbol{c}) \leq r$. Hence, $w(\boldsymbol{c}) = w(\boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{c}) \leq 2r + 1$. Therefore, if $w(\boldsymbol{c}) > 2r + 1$, then $\boldsymbol{c} \notin ZC$. □

THEOREM 4.6. *Let $\boldsymbol{c}_1$,$\boldsymbol{c}_2 \in \boldsymbol{C}$ and $\boldsymbol{c}_1$ be a proper descendant of $\boldsymbol{c}_2$. Then, $\boldsymbol{c}_2 \notin ZC$.*

Proof. Assume that $\boldsymbol{c}_2 \in ZC$ and $\boldsymbol{c}_3 = \boldsymbol{c}_1 \oplus \boldsymbol{c}_2$. Then, by Lemma 3.11, there exists an $\boldsymbol{x} \in F(\boldsymbol{0})$ such that $\boldsymbol{x} \in D_c(\boldsymbol{c}_2)$ and $\boldsymbol{x} \notin D_c(\boldsymbol{c}')$, $\boldsymbol{c}' \neq \boldsymbol{c}_2$, $\boldsymbol{c}' \in ZC$. Furthermore, by Lemma 3.6, $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}_2$. By Lemma 3.6, if $d(\boldsymbol{x}, \boldsymbol{c}_1) < w(\boldsymbol{x})$, then $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}_1$. In this case, $d(\boldsymbol{x}, \boldsymbol{c}_2) = d(\boldsymbol{x}, \boldsymbol{c}_1) + w(\boldsymbol{c}_3)$. Since $w(\boldsymbol{c}_3) \geq 2$, by Theorem 4.4, $\boldsymbol{x} \notin D_c(\boldsymbol{c}_2)$, which contradicts the statement that $\boldsymbol{x} \in D_c(\boldsymbol{c}_2)$. Therefore, $d(\boldsymbol{x}, \boldsymbol{c}_1) \geq w(\boldsymbol{x})$. Similarly, we have $d(\boldsymbol{x}, \boldsymbol{c}_3) \geq w(\boldsymbol{x})$. Therefore, $d(\boldsymbol{x}, \boldsymbol{c}_2) = d(\boldsymbol{x}, \boldsymbol{c}_1) + d(\boldsymbol{x}, \boldsymbol{c}_3) - w(\boldsymbol{x}) \geq w(\boldsymbol{x})$. This contradicts the statement that $\boldsymbol{x} \in D_c(\boldsymbol{c}_2)$. □

The above theorem is much less restrictive than Theorem 3 in [9] which states that if $\boldsymbol{c}_1$ and $\boldsymbol{c}_3$ are in N_0, then $\boldsymbol{c}_2 \notin N_0$. The following result gives a low bound on the number of codewords in ZC.

THEOREM 4.7. *All codewords of minimum weight belong to a ZC.*

Proof. Let $\boldsymbol{c}$ be a codeword of minimum weight. From Lemma 3.8, there exists one $\boldsymbol{x} \in F(\boldsymbol{0})$ and $\boldsymbol{x}$ is a descendant of $\boldsymbol{c}$. Thus, $d(\boldsymbol{c}, \boldsymbol{c}') \leq d(\boldsymbol{c}, \boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{c}') = w(\boldsymbol{c}) - w(\boldsymbol{x}) + d(\boldsymbol{x}, \boldsymbol{c}')$, where $\boldsymbol{c}' \neq \boldsymbol{c}$ and $\boldsymbol{c}' \in \boldsymbol{C}$. Hence, $d(\boldsymbol{x}, \boldsymbol{c}') \geq w(\boldsymbol{x}) + [d(\boldsymbol{c}, \boldsymbol{c}') - w(\boldsymbol{c})]$. Since $\boldsymbol{c}$ is of minimum weight, $d(\boldsymbol{c}, \boldsymbol{c}') - w(\boldsymbol{c}) \geq 0$. Thus, $d(\boldsymbol{x}, \boldsymbol{c}') \geq w(\boldsymbol{x})$. But since $\boldsymbol{x} \notin D(\boldsymbol{0})$, then $\boldsymbol{x} \in D_c(\boldsymbol{c})$, and $\boldsymbol{x} \notin D_c(\boldsymbol{c}')$. Therefore, $\boldsymbol{c} \in ZC$. □

5. Analysis of the size of a Zero-Coverings. In this section we give sizes of Zero-Coverings for some short codes that are obtained by computer searches. For long codes, an asymptotic bound on the size of a Zero-Coverings is given. As pointed out in section 2, the space and time complexity of the ZCA are determined by the size of a ZC. Therefore, we will focus on the discussion of the size of a ZC.

In Table 1 we give the sizes of the Zero-Coverings for some linear block codes. We also indicate, for comparison, the numbers of codewords and coset leaders for those codes. Since finding a ZC is an NP-hard computational problem (the minimum covering problem), for some codes we can obtain only upper bounds on the sizes of a ZC. The algorithm for solving the minimum covering problem used here is modified from the approximation algorithm given in [5].

TABLE 1
The sizes of Zero-Coverings for some linear block codes.

$code(n, k, d_{\min})$	2^k	2^{n-k}	$\|ZC\|$
$BCH(15, 7, 5)$	128	256	63
$QR(17, 9, 5)$	512	256	≤ 76
$BCH(21, 12, 5)$	4096	512	≤ 189
$QR(23, 11, 8)$	2048	4096	506
$QR(31, 16, 7)$	65536	32768	≤ 2271
$QR(47, 24, 11)$	16777216	8388608	≤ 17296

Now we give an asymptotic bound on the size of a ZC for long codes. The asymptotic bound will be characterized by the function

$$F_{ZC}(R) = \lim_{n\to\infty} 1/n \log_2 |ZC|,$$

where $R = k/n$ is the code rate [9].

THEOREM 5.1.

$$\begin{aligned} F_{ZC}(R) &\leq H_2(2H_2^{-1}(1-R)) - (1-R) \text{ when } R > 0.1887, \\ &\leq R \text{ otherwise,} \end{aligned}$$

where $H_2(x)$ is the binary entropy function of x and H_2^{-1} is the inverse of $H_2(x)$ for $0 \leq x \leq 1/2$.

Proof. For large n, the size of a ZC can be estimated by using the following facts:

1. The number of codewords with weight j, a_j can be estimated by $a_j = \binom{n}{j}/2^{n-k}$ for $j \geq d_{\min}$ [11].
2. For virtually all linear (n, k) codes,

$$r = nH_2^{-1}(1-R) + o(n),$$

where $o(n)$ denotes a function satisfying $\lim_{n\to\infty} o(n)/n = 0$ [4, 2, 8, 3].

3. For virtually all linear (n, k) codes, $d_{\min} \geq nH_2^{-1}(1-R) + o(n)$ [11, 3].

By Theorem 4.5 and fact 1, we have

$$|ZC| \leq \sum_{j=d_{\min}}^{2r+1} a_j.$$

By facts 2 and 3, the above inequality will be

$$|ZC| \leq (r+2)B,$$

where B is the largest value among $a_{d_{\min}}, a_{d_{\min}+1}, \ldots,$ and a_{2r+1}.

If $2r + 1 \leq \lfloor n/2 \rfloor$, then

$$B = a_{2r+1};$$

otherwise

$$B = \binom{n}{\lfloor n/2 \rfloor} / 2^{n-k}.$$

By calculation, when $R > 0.1887$, $2r+1 \le \lfloor n/2 \rfloor$, where $r = nH_2^{-1}(1-R)+o(n)$. Furthermore, by the relation

$$2^{nH_2(\lambda)-o(n)} \le \binom{n}{\lambda n} \le 2^{nH_2(\lambda)},$$

we have

$$\begin{aligned} B &= 2^{n[H_2(2H_2^{-1}(1-R))-(1-R)]} \text{ when } R > 0.1887, \\ &= 2^k \text{ otherwise.} \end{aligned}$$

Since $r + 2 = nH_2^{-1}(1-R) + o(n) + 2 \ll 2^k$ or $2^{n[H_2(2H_2^{-1}(1-R))-(1-R)]}$ when n is large, then

$$\begin{aligned} |ZC| &\le 2^{n[H_2(2H_2^{-1}(1-R))-(1-R)]} \text{ when } R > 0.1887, \\ &\le 2^k \text{ otherwise.} \end{aligned}$$

Therefore,

$$\begin{aligned} F_{ZC}(R) &\le H_2(2H_2^{-1}(1-R)) - (1-R) \text{ when } R > 0.1887, \\ &\le R \text{ otherwise.} \quad \square \end{aligned}$$

We remark here that the asymptotic bound turns out to be the same as that for the size of a Zero-Neighbors presented in [9] that is based on a geometric argument. However, the argument used here is simpler and more direct than that used in [9].

6. Conclusions. In this paper we have presented an improved ZNA-like decoding algorithm, the Zero-Coverings algorithm. The time and space complexity analysis of this algorithm are also given. Although the asymptotic bound given here indicates that the complexity of this algorithm is growing exponentially with code length n, from the computer simulation, a good computation gain can be obtained. For example, by the results in Table 1, the computation gain for code $(47, 24, 11)$ is at least $(2^{23}/17296)/24 = 20$. However, due to limitation of the memory and computation power of the computer, we can obtain simulation results only for short codes.

The decoding procedure presented here is a complete decoding procedure [11]. That is, the procedure always finds the codeword that is closest to the received vector. The procedure can be modified to an incomplete decoding (bounded-distance decoding) procedure in order to further reduce the decoding computation needed. Furthermore, although the decoding algorithm presented in this paper is designed for binary linear block codes, it can be generalized to nonbinary linear block codes.

Appendix. In this appendix we give an example to show that ZGA is not an optimal ZNA-like algorithm. Let code $(12, 5, 3)$ be a linear code generated by the following generating matrix:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

From computer simulation we have

$$ZG = ZC \cup \{111100000000\},$$

where ZC is a set containing 12 codewords. Thus, $|ZC|$ is less than $|ZG|$.

Acknowledgments. I would like to thank the reviewer for his invaluable suggestions, which I believe have helped me to improve the presentation of my paper. In addition, I would like to thank Elaine Weinman for her invaluable help on the language check.

REFERENCES

[1] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. Inform. Theory, IT-24 (1978), pp. 384–386.

[2] V. M. Blinovskii, *Lower asymptotic bound on the number of linear code words in a sphere of given radius in f_q^n*, Problemy Peredachi Informatsii, 23 (1987), pp. 50–53.

[3] J. T. Coffey and R. M. Goodman, *The complexity of information set decoding*, IEEE Trans. Inform. Theory, 36 (1990), pp. 1031–1037.

[4] G. Cohen, *A nonconstructive upper bound on covering radius*, IEEE Trans. Inform. Theory, 29 (1983), pp. 352–353.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1991.

[6] Y. S. Han and C. R. P. Hartmann, *The zero-guards algorithm for general minimum distance decoding problem*, IEEE Trans. Inform. Theory, 43 (1997), pp. 1655–1658.

[7] C. R. P. Hartmann and L. B. Levitin, *An improvement of the zero-neighbors minimum distance decoding algorithm: The zero-guard algorithm*, IEEE Internat. Symp. on Information Theory, Kobe, Japan, 1988.

[8] L. B. Levitin, *Covering radius of almost all linear codes satisfies the Goblick bound*, IEEE Internat. Symp. on Information Theory, Kobe, Japan, 1988.

[9] L. B. Levitin and C. R. P. Hartmann, *A new approach to the general minimum distance decoding problem: The zero-neighbors algorithm*, IEEE Trans. Inform. Theory, 31 (1985), pp. 378–384.

[10] L. B. Levitin, M. Naidjate, and C. R. P. Hartmann, *Generalized identity-guards algorithm for minimum distance decoding of group codes in metric space*, IEEE Internat. Symp. on Information Theory, San Diego, CA, 1990.

[11] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Elsevier, New York, 1977.

[12] W. W. Peterson, *Error-Correcting Codes*, 2nd ed., MIT Press, Cambridge, MA, 1972.